

Comparing definitions of weak higher categories, I

Dennis Borisov
Department of Mathematics, Yale University
dennis.borisov@gmail.com

September 14, 2009

Abstract

The theory of operads, defined through categories of labeled graphs, is generalized to suit definitions of higher categories with arbitrary basic shapes. Constructions of cubical, globular and opetopic weak higher categories are obtained as examples.

1 Introduction

This paper does *not* propose a new definition of weak higher categories. Here we develop the machinery (operads) for comparing higher categories, already defined using arbitrary combinatorics (globes, cubes, simplexes, opetopes etc.) and various types of structure weakening (contractibility, thin structure, slice construction, etc.).

Batanin in [Ba98] has introduced the notion of a globular operad, and used it to define weak globular categories, as representations of contractible globular operads. In this paper we describe how to define operads, starting not necessarily with globes, but with arbitrary shapes (for example cubes), and we obtain Batanin's construction as a particular case.

One of the reasons for success of the theory of usual operads (May, colored, cyclic, modular, PROPs, etc.), compared to the theory of monads for example, is the use of graphs. Here by a graph we mean the usual combinatorial object, given by vertices and edges.

However, the graphs used in definition of usual operads ([KM94], [BeM96], [GK98], [BoM08]), are not suitable for most theories of higher categories, where operations are also operands for other operations, and there is a hierarchy of dependencies. So if we want to extend the usual operadic theory

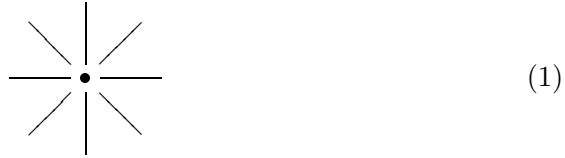
to include these dependencies we need to start with extending the notion of a graph.

There is an abundance of generalizations of graphs: poly-graphs, multi-graphs, hyper-graphs, and others, see for example [Bu93], [Pe99], [Ch06], [Ch03], [HMP98], [MT00], [Le04a]. However, most of the generalizations are either highly specialized for a particular type of combinatorics, or very general categorical constructions, that do not allow the same ease of manipulation, as the usual graphs.

In section 2 we provide, in a sense, “the minimal generalization” of the notion of graph, i.e. we add just enough structure to make it suitable for higher categories of any shape, and preserving ease of use.

Usual graphs are (relatively) easy to use, in part because the objects are relatively simple: intervals, connected at their ends. Our generalization is just as simple. In fact, we define generalized graphs to be very well known objects: direct categories. We call finite direct categories *nested graphs*, for reasons which will become clear in a moment.

Consider for example a usual corolla

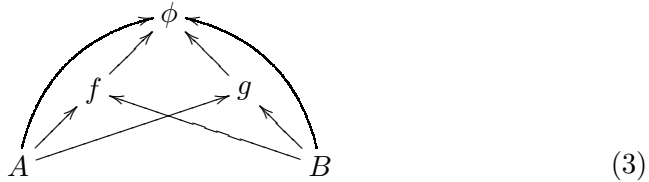


It consists of one vertex and several flags, attached to it. When we use (1) in the theory of usual operads, the vertex represents an operation, and ends of flags represent inputs. However, in the higher categorical context the inputs might be operations on their own, so they deserve their own vertices. Therefore, as a nested graph (1) is given by the following finite direct category



Nested corolla (2) does not carry more information than the plain one (1). The difference between plain and nested graphs really appears when we have at least three levels of dependencies. Consider the usual globular

2-morphism $A \begin{array}{c} \xrightarrow{f} \\ \Downarrow \phi \\ \xrightarrow{g} \end{array} B$. We can represent it as the following nested graph:



The morphism $A \rightarrow \phi$ is the composition $A \rightarrow f \rightarrow \phi$. This is a general principle: *if A is a flag of f and f is a flag of ϕ , then A is a flag of ϕ* , which we realize through the category structure on nested graphs. Here is the reason for the term “nested”: nested graphs are graphs decorated with other graphs, that are decorated with other graphs and so on, i.e. we have a nesting. Particular cases of nesting – trees decorated with trees – were considered by Batanin and Baez, Dolan.

Now we can create all kinds of compositions of operations, parameterized by nested graphs, that we obtain through gluing, just like with the plain graphs (i.e. non-nested ones). The compositions are then represented by functors between nested graphs, that contract or merge some of the flags or nodes.

In section 2 we build the double category of nested graphs. Plain graphs constitute an ordinary category (e.g. [BoM08]), where morphisms describe compositions. The case of nested graphs is more complicated, since in addition to compositions we have inclusions of one nested graph into another, and some of these inclusions correspond to dependencies between operations.

Therefore the double category of nested graphs \mathfrak{N} has a *category* of objects \mathfrak{O} and a *category* of morphisms \mathfrak{M} . The ordinary category of plain graphs \mathfrak{P} sits in \mathfrak{M} as a discrete double subcategory.¹

In section 3 we define operads, parameterized by nested graphs. We use the same approach, as was used in [BoM08] for usual graphs and operads. There it is noted that \mathfrak{P} is a monoidal category, with the product given by the direct sum, and \mathfrak{P} -operad in a given monoidal category \mathcal{M} is defined as a monoidal functor $\mathfrak{R} : \mathfrak{P} \rightarrow \mathcal{M}$, satisfying some conditions.

For every graph $\mathcal{G} \in \mathfrak{P}$ there is a morphism $\alpha : \coprod \mathcal{C}_i \rightarrow \mathcal{G}$ in \mathfrak{P} , where \mathcal{C}_i 's are the corollas in \mathcal{G} , and α is a grafting. Such α is called *an atom-*

¹Here and below, by a discrete double category we mean a double category having discrete categories of objects and morphisms.

ization morphism for \mathcal{G} , and the additional condition on \mathfrak{R} is that it maps atomization morphisms to isomorphisms.

The structure on \mathfrak{M} is more complicated than just a monoidal product. In a plain graph \mathcal{G} , vertices represent operations in an operad, and ends of flags represent elements in the algebra over this operad. Defining \mathfrak{R} we define only the operad, and not the algebra, i.e. we forget the ends of flags, and only choose operations for each vertex.

If \mathcal{H} is a nested graph, we cannot omit ends of flags, since they are operations too, and therefore giving the list of corollas in \mathcal{H} is not enough to determine the image of \mathcal{H} in \mathcal{M} (up to an isomorphism), i.e. monoidal structure is not enough for nested graphs.

To determine the image of \mathcal{H} in \mathcal{M} , we should give not only corollas, but also the diagram that glues them. The standard method for parameterizing gluings is using simplicial sets. We do this for \mathfrak{N} in section 3, and show that both \mathfrak{O} and \mathfrak{M} are algebras over a certain monad \mathfrak{S} , defined through simplicial sets, that describes gluings. If the simplicial sets are discrete we get the monoidal structure, given by the direct sum.

Then, as in [BoM08], we define the abstract categories of labeled nested graphs. These are actually double categories \mathfrak{G} , that have the \mathfrak{S} -algebra structure, together with an \mathfrak{S} -morphism $\mathfrak{G} \rightarrow \mathfrak{N}$. Representations of \mathfrak{G} are defined not in monoidal categories, but in \mathfrak{S} -algebras (every category with fiber products is automatically an \mathfrak{S} -algebra).

However, since the term “abstract category of labeled nested graphs” is rather long, we choose to call such \mathfrak{G} ’s *graph-algebras*.

In section 4 we provide two examples of our definitions. First we show that when we define a graph-algebra Φ , consisting of those nested graphs, that represent globular corollas and pasting diagrams, we get Batanin’s theory of globular operads and categories as representations of graph-algebras $\tilde{\Phi}$, having full and faithful morphism $\tilde{\Phi} \rightarrow \Phi$.

In the second example we define a graph-algebra Ξ , consisting of nested graphs that represent cubical corollas and diagrams, and we obtain cubical operads. It is known ([ABS02]), that strict globular and cubical categories are equivalent. In our language this can be stated as Φ, Ξ being Morita equivalent.

In fact, we construct a “Morita context” for this equivalence, i.e. a graph-algebra $\Phi + \Xi$, s.t. Φ and Ξ sit in it as sub-algebras. This construction allows us to translate globular operads and categories into cubical ones, and vice-versa, which will be done in [Bo09].

Choice of basic shapes is not the only choice, one has to make in developing a theory of higher categories. One also has to decide what are the *weak* categories, i.e. what structure weakening is allowed.

In [Ba98] a globular operad is called contractible if two different pastings for the same diagram are connected by higher morphisms. In our language this can be expressed as $\tilde{\Phi} \rightarrow \Phi$ having a right lifting property with respect to some double functors. This is similar to the notion of weak equivalence between simplicial categories.

A conceptually different approach is by using thin structure or slice construction, as discussed in [Str03], [BD98], [HMP00], [HMP01], [HMP02], [Ch04], [Hi05] and others.

Slice construction, as developed in [BD98], defines a new colored May operad P^+ for any colored May operad P . The idea is simple: compositions of operations in P should become operations in P^+ , and relations in P should become compositions in P^+ . Starting with the trivial operad, one iterates the slice construction to obtain a hierarchy of operations, that can be used to define weak higher categories.

In the last part of section 4 we give an outline of generalization of slice construction, applied to graph-algebras and operads defined by them. To keep the size of this paper within reasonable bounds, we postpone providing all the details until [Bo09].

In the context of nested graphs, slice construction becomes even more natural, than the original one. Indeed, a composition of operations in \mathfrak{G} is given by a functor $\phi : \mathcal{G}_1 \rightarrow \mathcal{G}_2$ between nested graphs. A straightforward procedure associates a nested graph \mathcal{G}_ϕ to ϕ , in fact, this is just a graph-theoretical version of constructing cone of a morphism in homological algebra.

In this way, for every $\phi \in \mathfrak{M}_{\mathfrak{G}}$ we add \mathcal{G}_ϕ to $\mathfrak{D}_{\mathfrak{G}}$. Given two composable ϕ, ψ , there is a natural gluing $\mathcal{G}_\phi * \mathcal{G}_\psi$, and the composition $\phi \circ \psi$ is given by a functor $\mathcal{G}_\phi * \mathcal{G}_\psi \rightarrow \mathcal{G}_{\phi \circ \psi}$, which we add to $\mathfrak{M}_{\mathfrak{G}}$.

However, $\phi \circ \psi$ are not all compositions we should add to $\mathfrak{M}_{\mathfrak{G}}$. And here we see an important difference between nested operads and colored May ones. Let A, B be two operations, and let $\{a_i\}, \{b_j\}$ be their flags. Since a_i 's and b_j 's are themselves operations, we can compose them, and then we should have a composition of A, B themselves.

In terms of the slice construction, this is expressed as follows: suppose we given four operations of composition $\phi, \psi, \alpha, \beta \in \mathfrak{M}_{\mathfrak{G}}$, making up a com-

mutative square

$$\begin{array}{ccc}
 \mathcal{G}_1 & \xrightarrow{\phi} & \mathcal{G}_2 \\
 \alpha \downarrow & & \downarrow \beta \\
 \mathcal{G}_3 & \xrightarrow{\psi} & \mathcal{G}_3
 \end{array} \tag{4}$$

We add the corresponding functor $(\alpha, \beta) : \mathcal{G}_\phi \rightarrow \mathcal{G}_\psi$ to $\mathfrak{M}_\mathfrak{G}$, and now we have the full slice construction \mathfrak{G}^+ .

When we apply the *full* slice construction iteratively to the trivial graph-algebra, we get not only the opetopes as in [BD98], but also horizontal products of globular corollas. In this way Batanin's theory and Baez-Dolan theory become parts of representations of one graph-algebra.

Contents

1	Introduction	1
2	The double category of nested graphs	7
2.1	The example of plain graphs	7
2.2	The category of objects	10
2.3	The category of morphisms	12
2.4	Compositions of morphisms	16
2.5	Associativity	20
3	Algebras of graphs and their representations	23
3.1	Monoidal categories revisited	23
3.2	Simplicial diagrams	28
3.3	Simplicial diagrams of nested graphs	34
3.4	Graph-algebras	40
4	Examples	44
4.1	Oriented graphs	46
4.2	Examples of oriented categories	50
4.2.1	Globes	50
4.2.2	Cubes	55
4.3	The slice construction and opetopes	56

2 The double category of nested graphs

In this section we construct the double category \mathfrak{N} of *nested graphs*. This double category will be used later to introduce nested operads, that are central in defining higher categories within any approach.

Later we will describe the algebraic structure, carried by \mathfrak{N} , and show it to be a generalization of the structure on (ordinary) category \mathfrak{P} of *plain graphs*, i.e. usual graphs that are not nested. This category is used (e.g. in [BoM08]) in definition of the usual (non-nested) operads, including May, cyclic, modular etc.

Therefore, we start this section with a description of \mathfrak{P} , and only after that we introduce the categories of objects and morphisms in \mathfrak{N} . Then we define the operation of composition on \mathfrak{M} , and prove that it makes \mathfrak{N} into a double category.

We define nested graphs as categories of a particular kind: *direct categories*. By definition, a **direct category** is a category \mathcal{G} , that admits a **grading**, i.e. a functor $\deg : \mathcal{G} \rightarrow \mathbf{n}$, where \mathbf{n} is an ordinal, considered as a category (with morphisms going from the smaller to the larger elements), and \deg sends non-identity morphisms to non-identity morphisms.² Plain graphs admit a grading into the ordinal $\mathbf{2}$, that has only two elements, i.e. in plain operads there are only two levels: operations and operands.

Clearly, a given direct category might have several, different gradings. We do not include a grading into the definition of a direct category, but only its existence. One can define finite direct categories in a different, but equivalent way, by requiring that they do not have non-trivial directed cycles of morphisms, i.e. composable sequences of non-identity morphisms, that start and end at the same object. This definition avoids the notion of a grading altogether. This is important since, in general, we do not require functors between direct categories to preserve gradings.

2.1 The example of plain graphs

Here we describe the category of plain graphs. The objects that we get are the same, as the graphs considered in [BoM08], but we arrive at them using direct categories. This will serve as a preparation for the next section, where we define general nested graphs.

Definition 1 A **plain graph** is a finite direct category $\deg : \mathcal{G} \rightarrow \mathbf{2}$, s.t.

²One can consider a more general definition, by allowing invertible morphisms to be sent to identity morphisms in \mathbf{n} , but we will not need it in this paper.

1. for any $A \in \deg^{-1}(0)$ there are at least 1 and at most 2 non-identity morphisms emanating from A ,
2. for any $B \in \deg^{-1}(1)$ there is at least one non-identity morphism coming into B .

In [BoM08], [BeM96], [GK98], [KM94] a finite graph was defined as a quadruple (V, F, ∂, j) , where V is the set of vertices, F is the set of flags, $\partial : F \rightarrow V$ is the attachment of flags to vertices, and $j : F \rightarrow F$ is an involution, connecting flags into edges, or leaving them alone as the legs of the graph.

Our definition translates into this one as follows: $V := \deg^{-1}(1)$, and $F \subset \text{Mor}(\mathcal{G})$ consisting of non-identity morphisms, $\partial : F \rightarrow \deg^{-1}(1)$ is the codomain map, and $j : F \hookrightarrow F$ is the involution, mapping each morphism to the other non-identity morphism, emanating from the same object.

In short, we encode vertices and ends of flags as objects, attachments of flags to vertices as morphisms, grafting of flags into an edge as two morphisms, coming from the same object.

Note that we disallow vertices, that do not have flags attached to them. Such graphs are allowed in [BoM08], and we could have included them here as well, but for simplicity of exposition we chose not to. They reappear again in the general theory of nested graphs.

Morphisms between plain graphs fall into 3 categories: graftings, mergers, contractions. *Graftings* identify ends of flags, which in our language means identifying objects in $\deg^{-1}(0)$, *mergers* identify vertices, i.e. objects in $\deg^{-1}(1)$. *Contractions* contract edges, which in our language means mapping non-identity morphisms to identities.

With an eye towards nested graphs, we will not differentiate between identifications of objects of different types, i.e. we will consider graftings and mergers as morphisms of one kind, and call all of them **mergers**.

Definition 2 A **merger** between two plain graphs is a surjective functor $\mu : \mathcal{G}_1 \rightarrow \mathcal{G}_2$, that is injective on non-identity morphisms, which means that if $\mu(f_1) = \mu(g_1)$ and $f_1 \neq g_1$, then both f_1 and g_1 are identities.

It is clear that fibers of a merger are discrete categories, which explains the meaning of the name: a merger can merge objects, but it does not identify different non-identity morphisms, nor does it map non-identity morphisms to identities.

It is easy to see, that a merger, according to our definition, is a mix of a merger and a grafting, as defined in [BoM08].

Contractions also may identify objects, but only if there is a sequence of non-identity morphisms between them, that are mapped to identity.

Definition 3 A contraction between plain graphs is a surjective functor $\kappa : \mathcal{G}_1 \rightarrow \mathcal{G}_2$, having the following properties:

1. fibers of κ are corollas, i.e. for any object $A_2 \in \mathcal{G}_2$, $\kappa^{-1}(A_2)$ has precisely one vertex,
2. κ contracts only entire edges, i.e. for any non-identity morphism $f \in \mathcal{G}_1$, s.t. $\kappa(f)$ is an identity, there is another non-identity morphism $g \in \mathcal{G}_1$, with the same domain, s.t. $\kappa(g)$ is the same identity.

Our definition of a contraction is narrower than the one in [BoM08], since we only allow contracting *loops*. However, it is easy to see that a contraction, as in [BoM08], is a composition of a merger and a contraction, as they are defined here.

Obviously the only functors that are simultaneously mergers and contractions are the isomorphisms.

Definition 4 Let $\mathcal{G}_1, \mathcal{G}_2$ be plain graphs. A morphism $\mathcal{G}_1 \Rightarrow \mathcal{G}_2$ is a pair of functors $\mathcal{G}_1 \xrightarrow{\mu} \xrightarrow{\kappa} \mathcal{G}_2$, where μ is a merger, and κ is a contraction.³

It is easy to see that given a functor $\mathcal{G}_1 \rightarrow \mathcal{G}_2$ there might be several (non-equivalent) ways to factorize this functor into a merger, followed by a contraction. These different choices correspond to specifying which of the objects, that get identified, are merged.

In [BoM08] this information is given by designating virtual edges. One can easily check that our definition of a morphism is equivalent to the one in [BoM08].

The way we defined them here, it is not clear, a priori, how to compose two morphisms between plain graphs. One has the obvious functorial composition: just compose the underlying functors.

Later (in the general framework of nested graphs) we will define compositions in the category of plain graphs as some special factorizations of such functorial compositions into mergers, followed by contractions. The resulting category is isomorphic to the category of graphs Gr in [BoM08].⁴

³To be precise we should define a morphism as an equivalence class of such pairs, where $(\mu_1, \kappa_1) \sim (\mu_2, \kappa_2)$, if there is an isomorphism α , s.t. $\mu_2 = \alpha\mu_1$ and $\kappa_2\alpha = \kappa_1$ as functors. For simplicity we will use representatives of these equivalence classes.

⁴Except for the exclusion of vertices without flags from our definition.

2.2 The category of objects

Here we generalize plain graphs from the previous section to graphs having a *recursive decoration*. By this we mean decorating flags of a graph with other graphs. Since decorations themselves can have decorations on their flags, and so on, we have a recursive process, that can be described by a direct category.

Definition 5 A nested graph \mathcal{G} is a finite direct category.

From now on all graphs that we will consider will be nested graphs (unless we explicitly specify them to be plain), and hence sometimes, instead of using the term nested graph, we will call them simply *graphs*.

Here are some terms we will be using, that will hopefully make the meaning of our constructions clearer.

Definition 6 • A **flag** in a nested graph is a non-identity morphism.

An **irreducible flag** is a flag, that cannot be written as a composition of two flags.

- Objects in a nested graphs will be called **nodes**. We consider every flag as being **decorated** by its domain. If a node is the codomain of a flag, we will say that **the flag is attached to the node**.
- A **vertex** in a graph is a node, that does not decorate any flag. A **corolla** is a nested graph with a unique vertex.
- A functor $\phi : \mathcal{G}_1 \rightarrow \mathcal{G}_2$ **contracts a flag** $f \in \text{Mor}(\mathcal{G}_1)$ if $\phi(f)$ is an identity. A **node A is contracted by a functor**, if it decorates a flag, that is contracted.
- For a functor $\phi : \mathcal{G}_1 \rightarrow \mathcal{G}_2$ between graphs, a **ϕ -fiber** is a subcategory of \mathcal{G}_1 of the form $\phi^{-1}(A_2)$ for some node $A_2 \in \mathcal{G}_2$. A **ϕ -vertex** is a vertex in a ϕ -fiber, considered as a graph on its own.

For plain graphs the correct notion of a sub-object⁵ is not just any subgraph, but a *full subgraph*, i.e. a subgraph that together with each vertex contains all of its legs. We have an analogous definition for nested graphs.

Definition 7 • A **full subgraph of a nested graph \mathcal{G}** is a subcategory $\mathcal{G}' \subseteq \mathcal{G}$, s.t. if a node belongs to \mathcal{G}' , so do all the flags attached to it.

⁵Correct from the point of view of the theory of operads.

- For a graph \mathcal{G} , a **\mathcal{G} -corolla** is a full subgraph of \mathcal{G} , that is a corolla. A **component of \mathcal{G}** is a maximal \mathcal{G} -corolla.

While plain graphs constitute a category, nested graphs are organized into a double category, i.e. instead of a set of objects in \mathfrak{P} , we have a category \mathfrak{D} of objects in \mathfrak{N} .

The reason for this is the nested structure of nested graphs, where operations not only act on their inputs, but also serve as operands for higher operations. In other words, we have a hierarchy of dependencies of operations on other operations. Hence the following definition.

Definition 8 The category of objects \mathfrak{D} has nested graphs as objects, and given two graphs \mathcal{G}, \mathcal{H} a morphism from \mathcal{G} to \mathcal{H} is an injective functor $\delta : \mathcal{H} \rightarrow \mathcal{G}$, s.t. the image $\delta(\mathcal{H})$ is a full subgraph of \mathcal{G} (Definition 7).

It is clear that composition of injective functors, having full subgraphs as images, is again injective and the image is a full subgraph, i.e. \mathfrak{D} is indeed a category.

In the next section we will define the category of morphisms in the double category of nested graphs, and objects in that category will be completely different from morphisms in \mathfrak{D} . To avoid terminological difficulties we will call morphisms in \mathfrak{D} **dependencies**. We will denote them by $\mathcal{G} > \mathcal{H}$.

Notice the change of direction: a dependency from \mathcal{G} to \mathcal{H} is given as a functor *to \mathcal{G} from \mathcal{H}* . This reversing of direction is needed, because later, defining higher categories, we will consider functors $\mathfrak{D} \rightarrow \text{Set}$, and images of dependencies will represent the source/target maps (generalized to \mathfrak{D} -combinatorics).

If both \mathcal{G} and \mathcal{H} are plain graphs, a dependency from \mathcal{G} to \mathcal{H} is just a usual subgraph $\mathcal{H} \hookrightarrow \mathcal{G}$, i.e. a union of corollas of \mathcal{G} . In particular, the only dependencies between plain corollas are isomorphisms.

We have defined direct categories as categories *admitting a grading*, but a choice of one is not part of the definition. However, sometimes it is useful to have a kind of “canonical” grading on a nested graph. There is one given by the height of nodes.

Definition 9 Let \mathcal{C} be a corolla. The **height** of \mathcal{C} is the length of a maximal string of composable flags in \mathcal{C} .

For a node A in an arbitrary nested graph \mathcal{G} , the **height** of A is the height of the \mathcal{G} -corolla, generated by A .

Clearly plain graphs are those graphs, all of whose vertices have height 1, and having some restrictions on valence of nodes. Graphs, all of whose vertices are of height 0, are just sets.

We have the following obvious result.

Proposition 1 *Let \mathcal{G} be a nested graph. For any node $A \in \mathcal{G}$, let $\text{height}(A)$ be the height of A . Then the map $A \mapsto \text{height}(A)$ is a grading on \mathcal{G} .*

Morphisms in \mathfrak{D} (Definition 8) preserve the grading by height.

2.3 The category of morphisms

As we have discussed in Section 2.1, a morphism between plain graphs consists of three parts: contracting flags, merging nodes and grafting flags. This triple structure is manifested by the fact that every such morphism can be uniquely factored into a grafting, followed by a merger, followed by a contraction ([BoM08]).

With nested graphs it is impossible to differentiate between a grafting and a merger, because every flag carries a decoration by a node, and hence every merger is a grafting and vice versa. Therefore we will call all such morphisms *mergers*.

Given two nested graphs $\mathcal{G}_1, \mathcal{G}_2$, a merger or a contraction $\mathcal{G}_1 \rightarrow \mathcal{G}_2$ is a functor satisfying certain conditions. There are two conditions, satisfied by morphisms of both types.

The first one is a kind of surjectivity. Since identifying objects in a category might lead to new compositions of morphisms, we do not require a literal surjectivity, but a categorical one: a functor $\phi : \mathcal{G}_1 \rightarrow \mathcal{G}_2$ is an **epi-functor** if the subcategory of \mathcal{G}_2 , generated by $\phi(\mathcal{G}_1)$, is all of \mathcal{G}_2 . It is clear that composition of epi-functors is an epi-functor, and invertible functors are epi-functors.

The second property, shared by both mergers and contractions, can be intuitively described as follows: any epi-functor $\phi : \mathcal{G}_1 \rightarrow \mathcal{G}_2$ describes compositions of operations, represented by nodes in \mathcal{G}_1 . If a node $A_1 \in \mathcal{G}_1$ is contracted by ϕ , it should affect all flags, decorated by A_1 , more precisely, every such flag should be either contracted, or identified with a flag, whose decoration is not contracted.

This property generalizes the condition, satisfied by contractions between plain graphs, that if half of an edge is contracted, the whole edge is contracted. Here is the precise definition.

Definition 10 *Let $\mathcal{G}_1, \mathcal{G}_2$ be graphs. An epi-functor $\phi : \mathcal{G}_1 \rightarrow \mathcal{G}_2$ is a **direct functor**, if for any flag $f_1 : A_1 \rightarrow B_1$ in \mathcal{G}_1 , that is not contracted by ϕ ,*

there is a ϕ -vertex A'_1 in $\phi^{-1}(\phi(A_1))$, and a flag f'_1 , decorated by A'_1 , s.t. $\phi(f'_1)$ factors $\phi(f_1)$.

Here by “factoring” we mean that there are two morphisms (not necessarily flags) $f_2, f'_2 \in \mathcal{G}_2$, s.t. $\phi(f_1) = f_2 \circ \phi(f'_1) \circ f'_2$. However, since A'_1 is supposed to be in the same ϕ -fiber as A_1 , and \mathcal{G}_2 is a direct category, it is clear that f'_2 has to be an identity.

These were the properties shared by mergers and contractions. The distinction between them comes from two types of elements in categories (and in nested graphs in particular): objects and morphisms.

Compositions of operations, represented by morphisms between graphs, involve identifications of objects and identifications of morphisms. We collect the former kind into the notion of a merger, and the latter – into the notion of a contraction.

- Definition 11** 1. A merger $\mu : \mathcal{G}_1 \rightarrow \mathcal{G}_2$ is a direct functor, that is injective on flags, i.e. it does not contract flags, and for two flags $f_1, g_1 \in \mathcal{G}_1$, if $f_1 \neq g_1$, then $\mu(f_1) \neq \mu(g_1)$.
2. A contraction $\kappa : \mathcal{G}_1 \rightarrow \mathcal{G}_2$ is a direct functor, s.t. for every node $A_2 \in \mathcal{G}_2$, the fiber $\kappa^{-1}(A_2)$ is a corolla (Definition 6).

Clearly fibers of mergers are discrete categories. Consequently, the only functors, that are simultaneously mergers and contractions, are invertible functors. Generalizing Definition 4 we define morphisms between nested graphs as follows.

Definition 12 A morphism between graphs $\mathcal{G}_1 \Rightarrow \mathcal{G}_3$ is a pair of functors

$$\mathcal{G}_1 \xrightarrow{\mu} \mathcal{G}_2 \xrightarrow{\kappa} \mathcal{G}_3 \quad (5)$$

where μ is a merger and κ is a contraction.⁶

To organize morphisms between graphs into a category \mathfrak{M} , we need to know what happens when we restrict $\mathcal{G}_1 \Rightarrow \mathcal{G}_2$ to full subgraphs $\mathcal{H}_1, \mathcal{H}_2$ of $\mathcal{G}_1, \mathcal{G}_2$.

Proposition 2 Let $\phi : \mathcal{G}_1 \rightarrow \mathcal{G}_2$ be a direct functor between graphs. Let $\mathcal{H}_2 < \mathcal{G}_2$ be a full subgraph, and let $\mathcal{H}_1 := \phi^{-1}(\mathcal{H}_2)$. Then the restriction $\phi' : \mathcal{H}_1 \rightarrow \mathcal{H}_2$ of ϕ is a direct functor as well.

⁶Below we will omit identities in designation of such pairs, i.e. instead of (μ, Id) we will write simply μ and so on.

Proof: It is clear that ϕ' is an epi-functor. Let A be a node in \mathcal{H}_1 , that is contracted by ϕ' , and let f be a flag in \mathcal{H}_1 , decorated by A , that is not contracted by ϕ' . Clearly A is contracted by ϕ , and f is not contracted by ϕ , therefore there is a flag $g \in \mathcal{G}_1$, decorated by a ϕ -vertex B , s.t. $\phi(g)$ factors $\phi(f)$. Since \mathcal{H}_2 is a full subgraph, and \mathcal{H}_1 is the inverse image of \mathcal{H}_2 , it is clear that $g \in \mathcal{H}_1$. It is also clear that $\phi'(g)$ factors $\phi'(f)$, and that B is a ϕ' -vertex. ■

Let $\mathcal{G}_1 \xrightarrow{\mu} \mathcal{G}_2 \xrightarrow{\kappa} \mathcal{G}_3$ be a morphism between graphs (Definition 12). Let $\mathcal{H}_3 < \mathcal{G}_3$ be a full subgraph. Applying proposition 2 twice we get the diagram

$$\begin{array}{ccccc} \mathcal{G}_1 & \xrightarrow{\mu} & \mathcal{G}_2 & \xrightarrow{\kappa} & \mathcal{G}_3 \\ \delta_1 \uparrow & & \delta_2 \uparrow & & \uparrow \delta_3 \\ \mathcal{H}_1 & \xrightarrow{\mu'} & \mathcal{H}_2 & \xrightarrow{\kappa'} & \mathcal{H}_3 \end{array} \quad (6)$$

and it is easy to see that κ' is a contraction and μ' is a merger. It is also obvious that $\delta_1, \delta_2, \delta_3$ are dependencies (i.e. morphisms in \mathfrak{D}). In such cases we will call the triple $(\delta_1, \delta_2, \delta_3)$ a **dependency** $(\mu, \kappa) > (\mu', \kappa')$.

Definition 13 The category of morphisms \mathfrak{M} is defined to have morphisms between nested graphs as objects, and dependencies between morphisms as morphisms.

As in the case of morphisms between plain graphs, there is some ambiguity, coming from the fact that invertible functors between nested graphs are both mergers and contractions.

Let $\mathcal{G}_1 \xrightarrow{\mu} \mathcal{G}_2 \xrightarrow{\kappa} \mathcal{G}_3$ be an object in \mathfrak{M} , and let $\mathcal{G}_1 \xrightarrow{\mu'} \mathcal{G}'_2 \xrightarrow{\kappa'} \mathcal{G}_3$ be another object, s.t. there is an invertible functor $\alpha : \mathcal{G}_2 \rightarrow \mathcal{G}'_2$ satisfying $\alpha \circ \mu = \mu'$, $\kappa = \kappa' \circ \alpha$.

These two objects are different, yet they carry the same algebraic information on compositions of operations, represented by nodes in \mathcal{G}_1 . However, we do not have to declare these objects of \mathfrak{M} equivalent, since the following diagram provides an isomorphism between them.

$$\begin{array}{ccccc} \mathcal{G}_1 & \xrightarrow{\mu} & \mathcal{G}_2 & \xrightarrow{\kappa} & \mathcal{G}_3 \\ \parallel & & \downarrow \alpha & & \parallel \\ \mathcal{G}_1 & \xrightarrow{\mu'} & \mathcal{G}'_2 & \xrightarrow{\kappa'} & \mathcal{G}_3 \end{array} \quad (7)$$

We have three functors

$$\sigma, \tau : \mathfrak{M} \rightarrow \mathfrak{D}, \quad \iota : \mathfrak{D} \rightarrow \mathfrak{M}, \quad (8)$$

sending (μ, κ) to \mathcal{G}_1 and \mathcal{G}_3 , and \mathcal{G} to $(Id_{\mathcal{G}}, Id_{\mathcal{G}})$ respectively. It is clear that they satisfy the usual identities of source, target, and identity automorphism maps. It remains to define associative compositions of objects in \mathfrak{M} . This will be done in the next two sections.

The following proposition expresses an important property of the category \mathfrak{M} : if we have a dependency $(\mu, \kappa) > (\mu', \kappa')$, then any composition of operations in the source of (μ', κ') , that is performed by (μ, κ) , is also performed by (μ', κ') . This property will be important in section 3.3, where we describe the algebraic structure, carried by the double category $(\mathfrak{D}, \mathfrak{M})$.

Proposition 3 *Let $(\delta_1, \delta_2, \delta_3) : (\mu, \kappa) > (\mu', \kappa')$ be a dependency as in (6). Let A be a node in \mathcal{H}_1 . If $\delta_1(A)$ is contracted by $\kappa \circ \mu$, then A is contracted by $\kappa' \circ \mu'$.*

Proof: Let f be a flag in \mathcal{G}_1 , that is decorated by $\delta_1(A)$, and is contracted by $\kappa \circ \mu$. Since the domain of f is in $\delta_1(\mathcal{H}_1)$, we have that $(\kappa \circ \mu)(f) \in \delta_3(\mathcal{H}_3)$, and therefore $f \in \delta_1(\mathcal{H}_1) = (\kappa \circ \mu)^{-1}(\delta_3(\mathcal{H}_3))$, which implies that $\kappa' \circ \mu'$ contracts $\delta_1^{-1}(f)$. ■

We finish this section with an example of something that our construction *does not* describe. We have defined objects of \mathfrak{M} as factorizations of direct functors into mergers, followed by contractions, and in this way we specify which nodes, that get contracted into the same node, are identified before the contraction. Consider for example the following direct functor

$$\begin{array}{ccc} & B & \xrightarrow{\quad} B \\ A \swarrow & & \nwarrow C \end{array} \quad (9)$$

Without a factorization into merger/contraction we cannot know if A, C are identified before being contracted into B .

However, there are pre-contraction identifications, that our definition does not isolate. Consider for example

$$A \rightrightarrows B \longrightarrow B. \quad (10)$$

There are two ways to perform this contraction: contract all flags at once, or identify the two flags, and then contract the result. Our definition of \mathfrak{M} does not let us distinguish between them.

2.4 Compositions of morphisms

In sections 2.2, 2.3 we have constructed two categories $\mathfrak{D}, \mathfrak{M}$, and two functors $\sigma, \tau : \mathfrak{M} \rightarrow \mathfrak{D}$. These categories represent objects and morphisms between nested graphs, and σ, τ are the source and the target maps respectively. In this section we define the functor of composition

$$\gamma : \mathfrak{M}_\sigma \prod_\tau \mathfrak{M} \rightarrow \mathfrak{M}. \quad (11)$$

Our definition of γ is based on composition of functors, however, since objects of \mathfrak{M} are not functors, but pairs of them, the construction is not immediate.

It is clear that compositions of mergers as functors are again mergers. The same claim about contractions is also true, but it is not obvious, and requires proof. We start with investigating the properties of direct functors.

Proposition 4 *Composition of two direct functors is direct.*

Proof: Let $\mathcal{G}_1 \xrightarrow{\phi_1} \mathcal{G}_2 \xrightarrow{\phi_2} \mathcal{G}_3$ be two direct functors between graphs. Let $f_1 : A_1 \rightarrow B_1$ be a flag in \mathcal{G}_1 , that is not contracted by $\phi_2 \circ \phi_1$. Clearly $\phi_1(f_1)$ is not identity and is not contracted by ϕ_2 . Therefore, there is a flag $f_2 \in \mathcal{G}_2$, decorated by a ϕ_2 -vertex A_2 in $\phi_2^{-1}(\phi_2(\phi_1(A_1)))$, s.t. $\phi_2(f_2)$ factors $\phi_2(\phi_1(f_1))$.

Since ϕ_1 is an epi-functor, we can write $f_2 = \phi_1(g_1^{(n)}) \circ \dots \circ \phi_1(g_1'') \circ \phi_1(g_1')$, for some flags $\{g_1^{(i)}\}$ in \mathcal{G}_1 , and we can assume these flags are not contracted by ϕ_1 . Therefore, there is a flag $f_1' \in \mathcal{G}_1$, decorated by a ϕ_1 -vertex A_1' in $\phi_1^{-1}(A_2)$, s.t. $\phi_1(f_1')$ factors $\phi_1(g_1')$. Clearly $\phi_2(\phi_1(f_1'))$ factors $\phi_2(\phi_1(f_1))$ and A_1' is a $\phi_2 \circ \phi_1$ -vertex in $(\phi_2 \circ \phi_1)^{-1}(\phi_2 \circ \phi_1)(A_1)$. ■

We have introduced the notion of direct functor between direct categories in order to generalize the property, that contractions between plain graphs have: an edge is either completely contracted, or left intact. The following two propositions explain why direct functors provide such a generalization.

Proposition 5 *Let $\phi : \mathcal{G}_1 \rightarrow \mathcal{G}_2$ be a direct functor. Any flag $f_2 \in \mathcal{G}_2$ can be factored into*

$$f_2 = \phi(f_1^{(n)}) \circ \dots \circ \phi(f_1'') \circ \phi(f_1'), \quad (12)$$

where $f_1^{(i)}$'s are flags between ϕ -vertices.

Proof: Since ϕ is an epi-functor, we can have a decomposition (12), but $f_1^{(i)}$'s are not necessarily flags between ϕ -vertices. Clearly we can assume that at least the codomains of $f_1^{(i)}$'s are ϕ -vertices, because we can compose them with flags, contracted by ϕ .

Consider f_1' . Since ϕ is direct, there is a flag g_1' , decorated by a ϕ -vertex, s.t. $\phi(g_1')$ factors $\phi(f_1')$. Therefore we can assume that f_1' itself is a flag between ϕ -vertices. Continuing like this, and taking into account that \mathcal{G}_1 is a finite category, we conclude that all $f_1^{(i)}$'s can be made into flags between ϕ -vertices. ■

Proposition 6 *Let $\kappa : \mathcal{G}_1 \rightarrow \mathcal{G}_2$ be a contraction. Then κ is surjective, and moreover, any flag in \mathcal{G}_2 can be lifted to \mathcal{G}_1 , so that both the domain and the codomain are κ -vertices.*

Proof: By Proposition 5, any flag $f_2 \in \mathcal{G}_2$ can be written as a composition of images of flags between κ -vertices in \mathcal{G}_1 . Since each κ -fiber has a unique vertex, such flags are composable in \mathcal{G}_1 , and the image of the composition is obviously f_2 . ■

Now we know enough about contractions to prove the following.

Proposition 7 *Composition of contractions is a contraction.*

Proof: Let $\mathcal{G}_1 \xrightarrow{\kappa_1} \mathcal{G}_2 \xrightarrow{\kappa_2} \mathcal{G}_3$ be a pair of contractions. By Proposition 4 all we have to prove is that $\kappa_2 \circ \kappa_1$ -fibers are corollas.

Let A_3 be a node in \mathcal{G}_3 . By assumption $\kappa_2^{-1}(A_3)$ is a corolla, let A_2 be the vertex in this corolla. By assumption $\kappa_1^{-1}(A_2)$ is also a corolla, and let A_1 be the vertex of this corolla. We claim that A_1 is the unique vertex in $(\kappa_2 \circ \kappa_1)^{-1}(A_3)$.

Let B_1 be a vertex in $(\kappa_2 \circ \kappa_1)^{-1}(A_3)$, it means that B_1 is not contracted by $\kappa_2 \circ \kappa_1$ into another node, therefore it is not contracted by κ_1 either, and hence B_1 is the unique vertex in $\kappa_1^{-1}(\kappa_1(B_1))$. If $\kappa_1(B_1) = A_2$, then $B_1 = A_1$, so suppose $\kappa_1(B_1) \neq A_2$.

Since $\kappa_2^{-1}(A_3)$ is a corolla, and $\kappa_1(B_1)$ is not its vertex, there is a flag f_2 , decorated by $\kappa_1(B_1)$, that is contracted by κ_2 . By Proposition 6 there is a flag f_1 in \mathcal{G}_1 , decorated by B_1 , s.t. $\kappa_1(f_1) = f_2$. But then $\kappa_2 \circ \kappa_1$ contracts f_1 – contradiction. So $B_1 = A_1$ and $(\kappa_2 \circ \kappa_1)^{-1}(A_3)$ is a corolla. ■

Since objects in \mathfrak{M} are pairs (μ, κ) , where μ is a merger, and κ is a contraction, to define compositions of such pairs we have to know how to compose mergers among themselves, contractions among themselves, and what is the composition of a contraction followed by a merger.

The case of mergers is easy, and Proposition 7 takes care of contractions. In the following definition we omit identities from pairs, that designate objects in \mathfrak{M} , i.e. instead of $(\mu, Id), (Id, \kappa)$ we write simply μ, κ .

Definition 14 *Let μ, μ' be mergers, and let κ, κ' be contractions, such that $\sigma(\mu) = \tau(\mu')$, $\sigma(\kappa) = \tau(\kappa')$. We define $\gamma(\mu, \mu')$, $\gamma(\kappa, \kappa')$ to be compositions of the underlying functors.*

The following proposition provides the main ingredient in the definition of composition of a contraction followed by a merger. In the proof of this proposition we construct a certain **minimal decomposition** of any direct functor into a merger, followed by a contraction. The word *minimal* refers to minimality of the merger in the decomposition.

Proposition 8 *Every direct functor can be written (not necessarily uniquely) as a merger, followed by a contraction.*

Proof: Let $\phi : \mathcal{G}_1 \rightarrow \mathcal{G}_2$ be a direct functor. There is an equivalence relation on the set of nodes in \mathcal{G}_1 , defined as follows: for $A_1 \neq B_1$, $A_1 \sim B_1$ if and only if both A_1 and B_1 are ϕ -vertices, and $\phi(A_1) = \phi(B_1)$. Let \mathcal{G}'_2 be the category, obtained by identifying equivalent nodes in \mathcal{G}_1 , with morphisms being freely generated by morphisms in \mathcal{G}_1 , subject to relations of composition in \mathcal{G}_1 .

There are two natural epi-functors $\mathcal{G}_1 \xrightarrow{\mu} \mathcal{G}'_2 \xrightarrow{\kappa} \mathcal{G}_2$. Indeed, μ is defined to send every object to its equivalence class, and every morphism to the corresponding generator in $Mor(\mathcal{G}'_2)$. Since we divided by relations of composition in \mathcal{G}_1 , μ is a functor.

Since \mathcal{G}'_2 is defined by generators and relations, it is enough to specify κ on objects and generators. Since $A_1 \sim B_1$ implies $\phi(A_1) = \phi(B_1)$, there is a well defined image in \mathcal{G}_2 for every object in \mathcal{G}'_2 . The generators of $Mor(\mathcal{G}'_2)$ are all morphisms in \mathcal{G}_1 , hence we can send them to their images under ϕ . Since relations are precisely all the compositions in \mathcal{G}_1 , we get a well defined functor $\kappa : \mathcal{G}'_2 \rightarrow \mathcal{G}_2$.

From the construction of μ, κ it is easy to see that

$$\phi = \kappa \circ \mu. \quad (13)$$

We claim that \mathcal{G}'_2 is a graph, μ is a merger, and κ is a contraction.

Suppose \mathcal{G}'_2 is not a direct category, i.e. there is a cycle of flags $f_2^{(n)} \xrightarrow{\quad} \dots \xrightarrow{\quad} f'_2$. Clearly μ is an epi-functor, therefore we can assume that

$$\forall i \quad f_2^{(i)} = \mu(f_1^{(i)}),$$

for some $f_1^{(i)} : A_1^{(i)} \rightarrow B_1^{(i)}$ in \mathcal{G}_1 . Since \mathcal{G}_1 is a direct category there is i , s.t. $B_1^{(i)} \neq A_1^{(i+1)}$. However, $\mu(B_1^{(i)}) = \mu(A_1^{(i+1)})$, because $f_2^{(i)}$ and $f_2^{(i+1)}$ are composable. Then, by definition of μ , $B_1^{(i)}$ and $A_1^{(i+1)}$ are ϕ -vertices, but $f_1^{(i+1)}$ is contracted by ϕ , since $f_2^{(i+1)}$ is contracted by κ (\mathcal{G}_2 is a direct category), so $A_1^{(i+1)}$ cannot be a vertex – contradiction.

Since μ is defined by identifying nodes, and it does not contract any flags, clearly it is a merger. Let $A_2 \in \mathcal{G}_2$ be a node. The fiber $\kappa^{-1}(A_2)$ has a unique vertex, because it is obtained by identifying all vertices in the fiber $(\mu \circ \kappa)^{-1}(A_2)$.

It remains to prove that κ is a direct functor. Let f_2 be a flag in \mathcal{G}_2' , that is not contracted by κ . Since μ is an epi-functor, we have

$$f_2 = \mu(f_1^{(n)}) \circ \dots \circ \mu(f_1'),$$

for some flags $f_1^{(i)} \in \mathcal{G}_1$. Let i be the smallest index, such that $\mu(f_1^{(i)})$ is not contracted by κ . Then (since ϕ is direct), there is a flag $g_1 \in \mathcal{G}_1$, s.t. it is decorated by a ϕ -vertex, and $\phi(g_1)$ factors $\phi(f_1^{(i)})$. Clearly $\mu(g_1)$ is not identity, it is decorated by a κ -vertex in the same κ -fiber, as the domain of f_2 , and $\kappa(\mu(g_1))$ factors $\kappa(f_2)$. ■

It is easy to see that, in general, a direct functor can be decomposed into a merger, followed by a contraction in more than one way. However, since mergers are epi-functors, if we fix the merger in the decomposition, the contraction is determined uniquely.

Definition 15 Let $\mathcal{G}_1 \xrightarrow{\kappa} \mathcal{G}_2$, $\mathcal{G}_2 \xrightarrow{\mu} \mathcal{G}_3$ be a contraction and a merger. Let ϕ be the composition of κ, μ as functors. We define $\gamma(\mu, \kappa)$ to be $\mathcal{G}_1 \xrightarrow{(\mu', \kappa')} \mathcal{G}_3$, where κ', μ' are obtained by factorization of ϕ as in (13).

Now, having two objects $(\mu, \kappa), (\mu', \kappa') \in \mathfrak{M}$ with matching source and target, we can define $\gamma((\mu', \kappa'), (\mu, \kappa))$ by first composing κ and μ' , and then the resulting pairs of mergers and contractions. In this way we get a map

$$\gamma : \text{Obj}(\mathfrak{M})_\sigma \prod_\tau \text{Obj}(\mathfrak{M}) \rightarrow \text{Obj}(\mathfrak{M}). \quad (14)$$

The following proposition extends this map to morphisms in \mathfrak{M} .

Here, for simplicity of exposition, we designate dependencies between objects in \mathfrak{M} as pairs of dependencies between graphs, rather than triples. It is easy to see that it does not create any ambiguity.

Proposition 9 Consider two morphisms in \mathfrak{M} as follows

$$\begin{array}{ccccccc}
 \mathcal{G}_1 & \xrightarrow{\mu_1} & & \xrightarrow{\kappa_1} & \mathcal{G}_2 & \xrightarrow{\mu_2} & \xrightarrow{\kappa_2} & \mathcal{G}_3 \\
 \delta_1 \uparrow & & & & \uparrow \delta_2 & & & \uparrow \delta_3 \\
 \mathcal{H}_1 & \xrightarrow{\mu'_1} & & \xrightarrow{\kappa'_1} & \mathcal{H}_2 & \xrightarrow{\mu'_2} & \xrightarrow{\kappa'_2} & \mathcal{H}_3
 \end{array} \tag{15}$$

Let $(\bar{\mu}, \bar{\kappa}) := \gamma((\mu_2, \kappa_2), (\mu_1, \kappa_1))$, $(\bar{\mu}', \bar{\kappa}') := \gamma((\mu'_2, \kappa'_2), (\mu'_1, \kappa'_1))$. Then

$$\begin{array}{ccc}
 \mathcal{G}_1 & \xrightarrow{\bar{\mu}} & \mathcal{G}_3 \\
 \delta_1 \uparrow & & \uparrow \delta_3 \\
 \mathcal{H}_1 & \xrightarrow{\bar{\mu}'} & \mathcal{H}_3
 \end{array} \tag{16}$$

is a morphism in \mathfrak{M}

Proof: The only condition from Definition 12 that is not clearly satisfied by (16) is that restriction of $(\bar{\mu}, \bar{\kappa})$ to the images of (δ_1, δ_3) equals $(\bar{\mu}', \bar{\kappa}')$. It is obvious that the underlying direct functors are equal, the question is about the factorization. To compare factorizations it is enough to compare the mergers, i.e. it is enough to prove that $\kappa_1 \circ \mu_1$ -vertices in the image of δ_1 are the same as $\kappa'_1 \circ \mu'_1$ -vertices. This follows from Proposition 3. ■

Proposition 10 The compositions from Definitions 14 and 15 extend to a functor

$$\mathfrak{M}_\sigma \prod_\tau \mathfrak{M} \rightarrow \mathfrak{M}. \tag{17}$$

Proof: This is almost obvious. A morphism in $\mathfrak{M}_\sigma \prod_\tau \mathfrak{M}$ can be described as a triple of dependencies $(\delta_1, \delta_2, \delta_3)$ (as in (15)), and we extend Definitions 14, 15 by sending $(\delta_1, \delta_2, \delta_3)$ to (δ_1, δ_3) . According to Proposition 9 we do get a morphism in \mathfrak{M} . Functoriality is obvious. ■

2.5 Associativity

In section 2.4 we have constructed a functor $\mathfrak{M}_\sigma \prod_\tau \mathfrak{M} \rightarrow \mathfrak{M}$. In section 2.3 we have defined $\iota : \mathfrak{D} \rightarrow \mathfrak{M}$ and $\sigma, \tau : \mathfrak{M} \rightarrow \mathfrak{D}$. Here we prove that $\mathfrak{N} := \{\mathfrak{D}, \mathfrak{M}, \iota, \sigma, \tau, \gamma\}$ is a double category. The only non-obvious property, that has not been proved yet, is associativity of γ .

By definition, γ consists of compositions of functors, and factorizations of direct functors into mergers, followed by contractions. Therefore, to prove

associativity of γ we should mostly analyze the factorizations, since the usual composition of functors is associative.

In the proof of the following proposition we do just that, and the reason why everything works out is that the factorization of direct functors, given by (13), into mergers, followed by contractions, is *minimal* among all possible factorizations.

Proposition 11 *Let $\gamma : \mathfrak{M}_\sigma \amalg_\tau \mathfrak{M} \rightarrow \mathfrak{M}$ be the functor from (17). Then the following diagram is commutative up to a unique natural isomorphism*

$$\begin{array}{ccc}
 \mathfrak{M}_\sigma \amalg_\tau \mathfrak{M}_\sigma \amalg_\tau \mathfrak{M} & \xrightarrow{\gamma \amalg Id} & \mathfrak{M}_\sigma \amalg_\tau \mathfrak{M} \\
 Id \amalg \gamma \downarrow & & \downarrow \gamma \\
 \mathfrak{M}_\sigma \amalg_\tau \mathfrak{M} & \xrightarrow{\gamma} & \mathfrak{M}
 \end{array} \tag{18}$$

Proof: Let $\mathcal{G}_1 \xRightarrow{(\mu_1, \kappa_1)} \mathcal{G}_2 \xRightarrow{(\mu_2, \kappa_2)} \mathcal{G}_3 \xRightarrow{(\mu_3, \kappa_3)} \mathcal{G}_4$ be 3 objects in \mathfrak{M} . We would like to prove that the two possible compositions coincide. Since composition of mergers is just composition of functors, it is easy to see that we can assume μ_1 to be trivial. Similarly we can assume κ_3 being trivial. Therefore, all we have to prove is that

$$\gamma(\mu_3, \gamma((\mu_2, \kappa_2), \kappa_1)) = \gamma(\gamma(\mu_3, (\mu_2, \kappa_2)), \kappa_1). \tag{19}$$

We will start with a simplification: let κ be a contraction, and let μ_1, μ_2 be mergers. We claim

$$\gamma(\mu_2 \mu_1, \kappa) = \gamma(\mu_2, \gamma(\mu_1, \kappa)). \tag{20}$$

Let's denote $\gamma(\mu_2 \mu_1, \kappa)$ by $((\mu_2 \mu_1)', \kappa')$, and $\gamma(\mu_2, \gamma(\mu_1, \kappa))$ by $(\mu_2' \mu_1', \kappa'')$. In construction of $(\mu_2 \mu_1)'$ we identify vertices in κ -fibers, over nodes, that are identified by $\mu_2 \mu_1$; on the right hand side this identification is split in two steps – first we identify according to μ_1 , and then μ_2 . Clearly $(\mu_2 \mu_1)' = \mu_2' \mu_1'$, and hence (20) is true, since any decomposition of a given direct functor into a merger, followed by a contraction, is determined by the merger.

Equation (20) implies that in (19) we can assume μ_2 to be trivial. And hence the problem is reduced to proving that for any contractions κ_1, κ_2 and a merger μ we have

$$\gamma(\mu, \kappa_2 \kappa_1) = \gamma(\gamma(\mu, \kappa_2), \kappa_1). \tag{21}$$

Let A be a node in the domain of μ , let B be the vertex in $\kappa_2^{-1}(A)$. The fiber $(\kappa_2\kappa_1)^{-1}(A)$ is a corolla, its vertex is the vertex C in $\kappa_1^{-1}(B)$. If μ identifies A and A' , then the merger μ' , obtained by commuting μ and $\kappa_2\kappa_1$, identifies C and C' .

On the right hand side we have two steps: first we identify B and B' , and then C and C' . The result is clearly the same, and so we obtain the same merger on both sides of (21), and therefore the same contraction. ■

Finally we can organize plain graphs into a category. Let $\mathfrak{D}_p, \mathfrak{M}_p$ be the discrete categories, obtained from $\mathfrak{D}, \mathfrak{M}$ by taking plain graphs and equivalence classes of contractions/mergers between them. Here we say that $(\mu, \kappa) \sim (\mu', \kappa')$ if there is an isomorphism α s.t.

$$\mu' = \alpha \circ \mu, \quad \kappa' \circ \alpha = \kappa. \quad (22)$$

Since $\{\mathfrak{D}_p, \mathfrak{M}_p, \sigma, \tau, \iota, \gamma\}$ is a discrete double category, it is just a category, and it is easy to see that it has the same objects and morphisms as \mathfrak{P} , defined in section 2.1.

Now we can also explain the reason for exclusion of the trivial graphs from \mathfrak{P} . Had we allowed corollas having no legs, we would not have been able to distinguish between such graphs and single flags, not attached to a node. The reason is obvious: since for us all nodes are operations, a point becomes a node, and not a flag, only when it has legs of its own.

So, by allowing trivial graphs in \mathfrak{P} , we would have obtained dependencies between objects of \mathfrak{P} , that are not inclusions of full plain subgraphs. Such dependencies are present in [BoM08], namely they are necessary to define algebras over operads. Indeed, an algebra is given by decorating legs of graphs with objects from a given category (usually the same object for all legs).

Here we have a very simple observation: in the theory of nested graphs and operads there are no algebras, only operads. In some cases one can artificially divide between algebras and operads. However, in general such divisions are impossible.

In [BoM08] all graphs have height ≤ 2 , and hence, differentiating by height, one can define operads and their algebras. In [Ba98] all graphs are oriented, and hence one can define endomorphism operads and representations.

In the rest of the paper, where we describe the algebraic structure, carried by \mathfrak{N} , it will be important for us to assume that $\mathfrak{D}, \mathfrak{M}$ are small categories. Since we have defined graphs as *finite* direct categories, we have

small versions of \mathfrak{D} , \mathfrak{M} . We will use the same notation \mathfrak{N} for such versions, and below we will always assume that $Obj(\mathfrak{D}), Obj(\mathfrak{M})$ are sets.

3 Algebras of graphs and their representations

In section 2 we have constructed a double category \mathfrak{N} of nested graphs, generalizing the category \mathfrak{P} of plain graphs. In this section we will use \mathfrak{N} to define higher (nested) operads, and eventually higher categories.

Our approach is an extension of the one, used in [BoM08], [BeM96], [GK98], [KM94] with \mathfrak{P} . There it is observed that \mathfrak{P} has a monoidal structure, given by the direct sum. This monoidal product allows us to decompose any plain graph into corollas. Then a \mathfrak{P} -operad in any monoidal category \mathcal{M} is just a monoidal functor $\mathfrak{P} \rightarrow \mathcal{M}$, that maps these decompositions into isomorphisms.

The case of \mathfrak{N} is more complicated, since operations have dependencies between them, and hence a monoidal structure is not enough to give decompositions of nested graphs into corollas. We have to take into account intersections of these corollas, intersections of the intersections and so on. Instead of a direct sum, a suitable operation is taking colimits of simplicial diagrams.

To have the right language, in this section we develop the machinery of simplicial sets, acting on categories, in the same way as actions of discrete simplicial sets produce monoidal structures. First we review how this approach works for the monoidal product, and then generalize it to arbitrary simplicial sets.

Then we use this language to describe the algebraic structure, present on \mathfrak{N} , and axiomatize it in the notion of a *graph-algebra*, which is the generalization to nested graphs of *abstract categories of labeled graphs* from [BoM08].

3.1 Monoidal categories revisited

Here we rewrite the definition of monoidal categories in a language that allows generalization to actions of arbitrary finite simplicial sets. We redefine monoidal categories as algebras over a monad \mathfrak{A} on Cat , constructed from a set of discrete finite categories $\{\mathcal{A}_n\}$.

After that, we recall the definition of operads from [BoM08], that uses monoidal structure on \mathfrak{P} , given by the direct sum.

Let \mathcal{A} be the subcategory of Cat , having $\{\mathcal{A}_n\}_{n \geq 0}$ as objects, where each \mathcal{A}_n is the ordinal \mathbf{n} , considered as a *discrete* category.⁷ The only morphisms in \mathcal{A} are the identity functors on \mathcal{A}_n 's.

Definition 16 Let \mathcal{M} be any category. We define $\mathfrak{A}(\mathcal{M})$ to be the category whose objects are functors $\mathcal{D}_n : \mathcal{A}_n \rightarrow \mathcal{M}$ for $n \geq 0$, and whose morphisms are commutative diagrams

$$\begin{array}{ccc} \mathcal{A}_m & \xrightarrow{F} & \mathcal{A}_n \\ & \searrow \mathcal{D}_m \quad \xrightarrow{\phi} \quad \searrow \mathcal{D}_n & \\ & \mathcal{M} & \end{array} \quad (23)$$

where F is a morphism in \mathcal{A} , and $\phi : \mathcal{D}_m \Rightarrow \mathcal{D}_n \circ F$ is a natural transformation.⁸

Clearly $\mathcal{M} \mapsto \mathfrak{A}(\mathcal{M})$ is a functor $\mathfrak{A} : Cat \rightarrow Cat$.

When $\mathcal{M} = \mathcal{A}$, there is a functor $\mathcal{K} : \mathfrak{A}(\mathcal{A}) \rightarrow \mathcal{A}$, defined by

$$\mathcal{K} : \mathcal{D}_m \mapsto \text{colim}(\mathcal{D}_m), \quad (24)$$

where colimit is taken in Cat , and then identified with an object in \mathcal{A} , preserving the (lexicographic) order on elements of \mathcal{A}_n 's.

The main property of \mathcal{K} is that for any category \mathcal{M} , it induces a functor

$$\mathcal{K}^* : \mathfrak{A}^2(\mathcal{M}) \rightarrow \mathfrak{A}(\mathcal{M}), \quad (25)$$

defined as follows: if $\mathcal{D}_m \in \mathfrak{A}^2(\mathcal{M})$ maps \mathcal{A}_m to $\{\mathcal{D}_{n_i}\}_{1 \leq i \leq m}$, then

$$\mathcal{K}^*(\mathcal{D}_m) := \coprod_{1 \leq i \leq m} \mathcal{D}_{n_i} : \mathcal{K}(\pi \circ \mathcal{D}_m) \rightarrow \mathcal{M}, \quad (26)$$

where $\pi : \mathfrak{A}(\mathcal{M}) \rightarrow \mathcal{A}$ sends \mathcal{D}_n to \mathcal{A}_n . Note that if $m = 0$, $\mathcal{K}^*(\mathcal{D}_m)$ is the unique functor $\mathcal{A}_0 \rightarrow \mathcal{M}$.

In addition, for any \mathcal{M} there is a trivial functor $\epsilon : \mathcal{M} \rightarrow \mathfrak{A}(\mathcal{M})$, sending each object to itself, considered as a diagram over \mathcal{A}_1 . We have the following obvious proposition.

⁷We need an order on the objects of \mathcal{A}_n 's because for a general monoidal product \otimes $A \otimes B$ is not the same as $B \otimes A$.

⁸Since we allow only identity morphisms in \mathcal{A} , it is clear that F has to be an identity, but, with a generalization below in mind, we prefer the more general formulation.

Proposition 12 *Defined as above, $\mathcal{K}^* : \mathfrak{A}^2 \rightarrow \mathfrak{A}$, $\epsilon : Id_{Cat} \rightarrow \mathfrak{A}$ are natural transformations.*

Less obvious is the following claim.

Proposition 13 *The triple $(\mathfrak{A}, \mathcal{K}^*, \epsilon)$ defines a monad on Cat .*

Proof: The statement is an easy consequence of commutativity of the following diagram

$$\begin{array}{ccc} \mathfrak{A}^2(\mathcal{A}) & \xrightarrow{\mathcal{K}^*} & \mathfrak{A}(\mathcal{A}) \\ \mathfrak{A}(\mathcal{K}) \downarrow & & \downarrow \mathcal{K} \\ \mathfrak{A}(\mathcal{A}) & \xrightarrow{\mathcal{K}} & \mathcal{A} \end{array} \quad (27)$$

which is equivalent to associativity of direct sum.⁹ ■

Definition 17 *An \mathfrak{A} -algebra is an algebra $(\mathcal{M}, \mathcal{R} : \mathfrak{A}(\mathcal{M}) \rightarrow \mathcal{M})$ over the monad $(\mathfrak{A}, \mathcal{K}^*, \epsilon)$.*

A morphism of \mathfrak{A} -algebras is a morphism of algebras over the monad \mathfrak{A} , with the usual diagrams commutative up to coherent natural isomorphisms.

It is easy to see that an \mathfrak{A} -algebra is just a strict monoidal category. Indeed, the functor

$$\mathfrak{A}(\mathcal{M}) \supseteq Fun(\mathcal{A}_2, \mathcal{M}) \rightarrow \mathcal{M} \quad (28)$$

defines the monoidal product. Its associativity is expressed by $\mathfrak{A}(\mathcal{M}) \rightarrow \mathcal{M}$ being an \mathfrak{A} -algebra, and the unit is given by the image of $\mathcal{A}_0 \rightarrow \mathcal{M}$ in \mathcal{M} .

Similarly a morphism of \mathfrak{A} -algebras is just a monoidal functor. We would get the general notion of a monoidal category, if we consider *pseudo*-algebras over \mathfrak{A} , but we do not need it in this paper, since the whole story of monoidal categories here is just an illustration, to prepare for the more complicated actions of simplicial sets, discussed in section 3.2.

A natural example of an \mathfrak{A} -category is any category \mathcal{M} , that has all finite direct sums, and where we define \mathcal{R} to map \mathcal{D}_n to $colim(\mathcal{D}_n)$. This is the case of the category \mathfrak{P} of plain graphs or of the category Cat of small categories.

⁹Since we have chosen special representatives for direct sums of objects in \mathcal{A} , the associativity here is on the nose, and not only up to coherent natural isomorphisms.

In order to produce a meaningful theory of (plain) operads, sometimes one has to consider only parts of \mathfrak{P} , or, on the other hand, plain graphs with additional structure (labeling), and so on. To formalize many of these constructions, we need to axiomatize properties of \mathfrak{P} , that we would like abstract categories of labeled graphs also to have.

Here are the properties of \mathfrak{P} , that we would like to axiomatize:

1. It is an \mathfrak{A} -algebra (i.e. a monoidal category).
2. It has a (full) subcategory \mathfrak{P}_1 of corollas, s.t. for every morphism $\phi : \mathcal{G}' \rightarrow \mathcal{G}$ in \mathfrak{P} , there is a unique (up to a unique isomorphism, trivial on ϕ) diagram

$$\begin{array}{ccc} \coprod \mathcal{G}'_i & \xrightarrow{\quad} & \coprod \mathcal{C}_i \\ \alpha' \downarrow & & \downarrow \alpha \\ \mathcal{G}' & \xrightarrow{\quad \phi \quad} & \mathcal{G} \end{array} \quad (29)$$

where each \mathcal{C}_i is a corolla, α, α' are mergers, and are injective, when restricted to $\mathcal{G}'_i, \mathcal{C}_i$ respectively; and for every object $\mathcal{G} \in \mathfrak{P}$ there is at least one morphism

$$\beta : \mathcal{G} \rightarrow \mathcal{C}, \quad (30)$$

where \mathcal{C} is a corolla.

We will call diagram (29) **the atomization diagram for ϕ** , and α will be called **the atomization morphism** for \mathcal{G} , we will call β from (30) **a full contraction of \mathcal{G}** .

3. For any set of full contractions $\{\phi_i : \mathcal{G}'_i \rightarrow \mathcal{C}_i\}$ and an atomization morphism $\alpha : \coprod \mathcal{C}_i \rightarrow \mathcal{G}$, there is a unique (up to a unique isomorphism) morphism $\phi : \mathcal{G}' \rightarrow \mathcal{G}$, s.t. its atomization diagram is

$$\begin{array}{ccc} \coprod \mathcal{G}'_i & \xrightarrow{\coprod \phi_i} & \coprod \mathcal{C}_i \\ \downarrow & & \downarrow \alpha \\ \mathcal{G}' & \xrightarrow{\quad \phi \quad} & \mathcal{G} \end{array} \quad (31)$$

In short, \mathfrak{P} is an \mathfrak{A} -algebra, that is generated by corollas, i.e. every object can be decomposed into corollas, and every morphism can be decomposed into morphisms with corollas as codomains, and these decompositions are compatible with compositions of morphisms. Axiomatically this can be stated as follows.

Definition 18 ([BoM08]) **An abstract category of labeled plain graphs** is an \mathfrak{A} -algebra (\mathfrak{L}, \otimes) , and a faithful monoidal functor $U : \mathfrak{L} \rightarrow \mathfrak{P}$, s.t.

1. For every object $\mathcal{L} \in \mathfrak{L}$ there is at least one morphism

$$\beta : \mathcal{L} \rightarrow \mathcal{C}, \quad (32)$$

s.t. $U(\beta)$ is a full contraction.

2. For every morphism $\phi : \mathcal{L}' \rightarrow \mathcal{L}$ in \mathfrak{L} , there is a unique (up to unique isomorphism, trivial on ϕ) diagram

$$\begin{array}{ccc} \bigotimes \mathcal{L}'_i & \xrightarrow{\quad} & \bigotimes \mathcal{C}_i \\ \alpha' \downarrow & & \downarrow \alpha \\ \mathcal{L}' & \xrightarrow[\phi]{} & \mathcal{L} \end{array} \quad (33)$$

where each $U(\mathcal{C}_i)$ is a corolla, and the image of U , applied to the diagram above, is the atomization of $U(\phi)$. We will call this diagram **the atomization diagram for ϕ** , and α will be called **the atomization morphism for \mathcal{L}** .

3. For any set of morphisms $\{\phi_i : \mathcal{L}'_i \rightarrow \mathcal{C}_i\}$ and an atomization morphism $\alpha : \bigotimes \mathcal{C}_i \rightarrow \mathcal{L}$, there is a unique (up to a unique isomorphism) morphism $\phi : \mathcal{L}' \rightarrow \mathcal{L}$, s.t. its atomization diagram is

$$\begin{array}{ccc} \bigotimes \mathcal{L}'_i & \xrightarrow[\bigotimes \phi_i]{} & \bigotimes \mathcal{C}_i \\ \downarrow & & \downarrow \\ \mathcal{L}' & \xrightarrow[\phi]{} & \mathcal{L} \end{array} \quad (34)$$

The definition of an abstract category of labeled plain graphs \mathfrak{L} is the basis for the theory of (plain) operads. Let \mathcal{M} be a monoidal category. **An \mathfrak{L} -operad in \mathcal{M}** is a morphism of \mathfrak{A} -algebras $\mathfrak{L} \rightarrow \mathcal{M}$ (i.e. a monoidal functor), that maps atomization morphisms to isomorphisms.

In the next two sections we are going to define another monad \mathfrak{S} on Cat , given by arbitrary finite simplicial sets. When restricted to discrete finite simplicial sets \mathfrak{S} becomes essentially \mathfrak{A} .

However, when the simplicial sets are not discrete, and \mathfrak{S} is acting on a non-discrete category, the resulting structure is more complicated, than just a monoidal product.

We will define an action of \mathfrak{S} on the double category \mathfrak{N} of nested graphs, and use it to introduce the notion of *graph-algebras*, which are generalizations of abstract categories of labeled plain graphs, and provide the basis for the theory of nested operads.

3.2 Simplicial diagrams

In section 3.1 we have re-formulated the notion of monoidal categories in terms of a particular monad \mathfrak{A} on Cat . Given a category \mathcal{M} , an action of \mathfrak{A} on \mathcal{M} is defined by functorially assigning an object of \mathcal{M} to every functor $\mathcal{A}_n \rightarrow \mathcal{M}$, where \mathcal{A}_n is a discrete category with n objects.

In this section we use the same approach – assignment of objects to diagrams, but we will use diagrams, parameterized by categories that are not discrete. More precisely, we will use finite simplicial sets to parameterize diagrams.

Let $SSet$ be the category of *finite* simplicial sets. We will consider it as a subcategory of Cat , where we think of a simplicial set $\mathcal{S} = \{S_k\}_{k \geq 0}$ as a category by taking $\coprod_{k \geq 0} S_k$ as the set of objects, and the morphisms being generated by the structure maps $\{\partial_i : S_k \rightarrow S_{k-1}\}$, $\{d_j : S_k \rightarrow S_{k+1}\}$, subject to the usual simplicial relations. A morphism of simplicial sets maps structure maps to structure maps, and hence it induces a functor between the corresponding categories.

Definition 19 *Let \mathcal{M} be any category. We define $\mathfrak{S}(\mathcal{M})$ to be the category, whose objects are functors $D_{\mathcal{S}} : \mathcal{S} \rightarrow \mathcal{M}$ for some $\mathcal{S} \in SSet$, and morphisms are commutative diagrams*

$$\begin{array}{ccc}
 \mathcal{S} & \xrightarrow{F} & \mathcal{T} \\
 \searrow \scriptstyle D_{\mathcal{S}} & \xRightarrow{\phi} & \nearrow \scriptstyle D_{\mathcal{T}} \\
 & \mathcal{M} &
 \end{array} \tag{35}$$

where $F : \mathcal{S} \rightarrow \mathcal{T}$ is a morphism in $SSet$, and $\phi : D_{\mathcal{S}} \Rightarrow D_{\mathcal{T}} \circ F$ is a natural transformation.

It is clear that $\mathcal{M} \mapsto \mathfrak{S}(\mathcal{M})$ is a functor $\mathfrak{S} : Cat \rightarrow Cat$. There is also an obvious projection

$$\pi : \mathfrak{S}(\mathcal{M}) \rightarrow SSet, \tag{36}$$

sending \mathcal{D}_S to \mathcal{S} , and (F, ϕ) to F .

When $\mathcal{M} = SSet$, there is a functor $\mathcal{K} : \mathfrak{S}(SSet) \rightarrow SSet$, defined by

$$\mathcal{K} : \mathcal{D}_S \mapsto \text{diag}(\coprod \mathcal{D}_S), \quad (37)$$

where $\coprod \mathcal{D}_S$ is the bisimplicial set, obtained from \mathcal{D}_S by taking direct sums of simplicial sets in the image of each S_k ; and diag is the diagonal functor from bisimplicial sets to simplicial sets.

Let \mathcal{M} be any category. We define a functor

$$\mathcal{K}^* : \mathfrak{S}^2(\mathcal{M}) \rightarrow \mathfrak{S}(\mathcal{M}), \quad (38)$$

as follows. Let $\mathcal{S} \in SSet$, and let $\mathcal{D}_S : \mathcal{S} \rightarrow \mathfrak{S}(\mathcal{M})$ be an object in $\mathfrak{S}^2(\mathcal{M})$. We have a simplicial set $\mathcal{K}(\pi \circ \mathcal{D}_S)$, where π is the projection from (36). Since this simplicial set is obtained by taking a direct sum and a diagonal, there is a functor

$$\mathcal{K}^*(\mathcal{D}_S) : \mathcal{K}(\pi \circ \mathcal{D}_S) \rightarrow \mathcal{M}, \quad (39)$$

defined by \mathcal{D}_S .

For any category \mathcal{M} we also have a functor $\epsilon_{\mathcal{M}} : \mathcal{M} \rightarrow \mathfrak{S}(\mathcal{M})$, sending every object $A \in \mathcal{M}$ to $\mathcal{D}_{pt} : pt \mapsto A$, where pt is the one-point simplicial set.

Proposition 14 *As defined above, $\epsilon : Id_{Cat} \rightarrow \mathfrak{S}$, $\mathcal{K}^* : \mathfrak{S}^2 \rightarrow \mathfrak{S}$ are natural transformations.*

Proof: Straightforward application of definitions. ■

Proposition 15 *The triple $(\mathfrak{S}, \mathcal{K}^*, \epsilon)$ defines a 2-monad on Cat .*

Proof: We claim that \mathfrak{S} is only weakly associative (i.e. a 2-monad) and not strictly, because direct sum in Set (and therefore in $SSet$) is not strictly associative, but only up to a unique isomorphism. However, this universal property of direct sums implies that we do not have to check the coherence conditions for the natural isomorphisms, and we can assume in this proof that the direct sums in Set are associative on the nose.

The claim of the Proposition is a consequence of commutativity of the following diagram:

$$\begin{array}{ccc} \mathfrak{S}^2(SSet) & \xrightarrow{\mathcal{K}^*} & \mathfrak{S}(SSet) \\ \mathfrak{S}(\mathcal{K}) \downarrow & & \downarrow \mathcal{K} \\ \mathfrak{S}(SSet) & \xrightarrow{\mathcal{K}} & SSet \end{array} \quad (40)$$

To equate the two possible functors from $\mathfrak{S}^2(SSet)$ to $SSet$ in (40), we factorize each one of them in a different way, and equate the factorizations.

Consider the following two operations, performed on $\mathfrak{S}^2(SSet)$: the first one is building simplicial diagrams in bisimplicial sets and then continuing to trisimplicial sets; the second one is building bisimplicial diagrams in simplicial sets, and then continuing to the trisimplicial sets. These constructions lead to the same result (up to a unique natural isomorphism), since taking direct sums is an associative procedure.

Now having a trisimplicial set we can take the diagonal in two ways, which are equal. It is easy to see that the two functors in (40) are obtained by building trisimplicial sets in two ways, and then taking the diagonal in two ways. Therefore, the result is the same. ■

In the following definition we use algebras over 2-monads. As with algebras over any kind of monads, there are some diagrams that have to be commutative. In case of a 2-monad, the commutativity is in general only up to a natural transformation ([KS74]). We would like to require that these natural transformations are invertible.

Definition 20 *An \mathfrak{S} -algebra is an algebra $(\mathcal{M}, \mathcal{R} : \mathfrak{S}(\mathcal{M}) \rightarrow \mathcal{M})$ over the 2-monad $(\mathfrak{S}, \mathcal{K}^*, \epsilon)$, s.t. the usual diagrams are commutative up to natural isomorphisms, that satisfy coherence conditions.*

A morphism between \mathfrak{S} -algebras *is a morphism between algebras over the 2-monad, s.t. the usual diagrams are commutative up to natural isomorphisms, that satisfy coherence conditions.*

Consider an example of \mathfrak{S} -algebra $(\mathcal{M}, \mathcal{R})$, s.t. \mathcal{M} is a discrete category. It is clear, that in this case any functor $\mathcal{S} \rightarrow \mathcal{M}$, where \mathcal{S} is a simplicial set, factors through \mathcal{A}_n , where n is the number of connected components in \mathcal{S} , and \mathcal{A}_n is a discrete category with n -objects. Therefore, an \mathfrak{S} -algebra structure on a discrete category carries as much information as a (symmetric) monoidal structure.

The main tool in this paper is construction of an \mathfrak{S} -algebra structure on the double category \mathfrak{N} , i.e. both the category of objects \mathfrak{D} , and the category of morphisms \mathfrak{M} are \mathfrak{S} -algebras in a compatible way. Inside these categories we have the discrete categories \mathfrak{D}_p , \mathfrak{M}_p of plain graphs and morphisms between them, and we will see that properties of \mathfrak{P} , that we have axiomatized in section 3.1 are degenerations of properties of \mathfrak{N} .

The following proposition provides the main mechanism for constructing \mathfrak{S} -algebras. It states that any finitely cocomplete category is automatically an \mathfrak{S} -algebra, and hence we get many more by taking its subcategories. For

our purposes here, the two most important examples of finitely cocomplete categories are *Cat* and *Set*.

Proposition 16 *Let \mathcal{M} be a category having all finite colimits. For any $\mathcal{D}_S \in \mathfrak{S}(\mathcal{M})$ let $\mathcal{R}(\mathcal{D}_S)$ be the colimit of \mathcal{D}_S . Then $(\mathcal{M}, \mathcal{R})$ is an \mathfrak{S} -algebra.*

Proof: It is easy to see that \mathcal{R} is a functor. We need to prove that the diagram

$$\begin{array}{ccc} \mathfrak{S}^2(\mathcal{M}) & \xrightarrow{\mathfrak{S}(\mathcal{R})} & \mathfrak{S}(\mathcal{M}) \\ \mathcal{K}^* \downarrow & & \downarrow \mathcal{R} \\ \mathfrak{S}(\mathcal{M}) & \xrightarrow{\mathcal{R}} & \mathcal{M} \end{array} \quad (41)$$

is commutative up to a coherent natural isomorphism.

It is easy to see that for any $\mathcal{D}_S \in \mathfrak{S}^2(\mathcal{M})$, the object $(\mathcal{R} \circ \mathfrak{S}(\mathcal{R}))(\mathcal{D}_S)$ is a colimit of the diagram $\mathcal{K}(\pi \circ \mathcal{D}_S) \rightarrow \mathcal{M}$. On the other hand, a standard argument shows that colimit of a diagram, parameterized by a bisimplicial set, is also a colimit of the simplicial diagram, obtained by taking the diagonal. The latter is precisely $(\mathcal{R} \circ \mathcal{K}^*)(\mathcal{D}_S)$.

Since this isomorphism $\mathcal{R} \circ \mathcal{K}^* \simeq \mathcal{R} \circ \mathfrak{S}(\mathcal{R})$ is obtained by using universal properties of colimits, coherence conditions are automatically satisfied. ■

Since we will be often using double categories, that carry an \mathfrak{S} -algebra structure on both the category of objects and the category of morphisms, in a compatible way, we need a special term for such double categories.

Recall that a double category is given by $(\mathcal{O}, \mathcal{M}, \sigma, \tau, \iota, \gamma)$, where \mathcal{O}, \mathcal{M} are categories, $\sigma, \tau : \mathcal{M} \rightarrow \mathcal{O}$, $\mathcal{O} \rightarrow \mathcal{M}$ are source, target, and identity maps respectively, and $\gamma : \mathcal{M}_\sigma \amalg_\tau \mathcal{M} \rightarrow \mathcal{M}$ is the composition map, s.t. together they constitute a category object in *Cat*.

Definition 21 *An \mathfrak{S} -category is a double category $(\mathcal{O}, \mathcal{M}, \sigma, \tau, \iota, \gamma)$, s.t. both \mathcal{O} and \mathcal{M} are \mathfrak{S} -algebras, in a compatible way, i.e. σ, τ, ι are morphisms of \mathfrak{S} -algebras, and the following diagram is commutative (up to coherent natural isomorphisms):*

$$\begin{array}{ccc} \mathfrak{S}(\mathcal{M})_{\mathfrak{S}(\sigma)} \amalg_{\mathfrak{S}(\tau)} \mathfrak{S}(\mathcal{M}) & \xrightarrow{\mathfrak{S}(\gamma)} & \mathfrak{S}(\mathcal{M}) \\ \mathcal{R} \amalg \mathcal{R} \downarrow & & \downarrow \mathcal{R} \\ \mathcal{M}_\sigma \amalg_\tau \mathcal{M} & \xrightarrow{\gamma} & \mathcal{M} \end{array} \quad (42)$$

Usually we will denote a double category by the categories of objects and morphisms only. An important example of an \mathfrak{S} -category is provided by any \mathfrak{S} -algebra, if we consider morphisms as objects.

Proposition 17 *Let $(\mathcal{M}, \mathcal{R})$ be an \mathfrak{S} -algebra. Let $(\mathcal{M}, \text{Mor}(\mathcal{M}))$ be the double category, where objects in $\text{Mor}(\mathcal{M})$ are morphisms in \mathcal{M} , and morphisms are commutative squares. Then $(\mathcal{M}, \text{Mor}(\mathcal{M}))$ is an \mathfrak{S} -category.*

Proof: Straightforward. It is an easy consequence of \mathcal{R} being a functor. ■

As with any algebraic object, admitting a free construction, one can introduce the notion of generators for an \mathfrak{S} -algebra \mathcal{M} . This would be a subcategory $\mathcal{N} \subseteq \mathcal{M}$, s.t. every object and every morphism in \mathcal{M} can be written using \mathfrak{S} -products and compositions of objects, morphisms in \mathcal{N} .

In this paper, however, we will use the term *generators* of an \mathfrak{S} -algebra in a different way, that is more suitable for applications to the theory of operads.

Definition 22 *Let \mathcal{M} be an \mathfrak{S} -algebra. We will say that \mathcal{M} is **generated** by a set $G \subseteq \text{Obj}(\mathcal{M})$, if every object in \mathcal{M} can be written as an \mathfrak{S} -product of elements of G , and every morphisms in \mathcal{M} can be obtained by taking \mathfrak{S} -products and compositions of morphisms, having elements of G as codomains.*

*We will say that an \mathfrak{S} -category $(\mathcal{O}, \mathcal{M})$ is **generated** by a set of objects $\mathfrak{C} \subseteq \text{Obj}(\mathcal{O})$, if the \mathfrak{S} -algebra \mathcal{O} is generated by \mathfrak{C} , and the \mathfrak{S} -algebra \mathcal{M} is generated by $\tau^{-1}(\mathfrak{C}) \subseteq \text{Obj}(\mathcal{M})$.¹⁰*

To see why this definition of generation is more suitable from the point of view of operads, consider the example of an \mathfrak{S} -category, that is discrete as a double category (i.e. both \mathcal{M}, \mathcal{O} are discrete categories). Then \mathfrak{S} -structure is reduced to a monoidal product, and if $(\mathcal{O}, \mathcal{M})$ is generated by $\mathfrak{C} \subseteq \text{Obj}(\mathcal{O})$, then $(\mathcal{O}, \mathcal{M})$ is just a colored May operad, with the set of colors being \mathfrak{C} .¹¹

We will define an \mathfrak{S} -algebra structure on the double category of nested graphs $\mathfrak{N} = (\mathfrak{D}, \mathfrak{M})$ by taking colimits in *Cat*. We will see that colimits of objects in $\mathfrak{S}(\mathfrak{D})$ are themselves nested graphs. However, not all morphisms in $\mathfrak{S}(\mathfrak{D})$ are mapped by *colim* to dependencies between graphs, and similarly for \mathfrak{M} . In the next section, using the notion of *fibrations* between objects in $\mathfrak{S}(\mathfrak{D})$, we will single out those, that are mapped by *colim* to dependencies.

¹⁰Recall that $\tau : \mathcal{M} \rightarrow \mathcal{O}$ is the target map of the double category $(\mathcal{O}, \mathcal{M})$.

¹¹To be precise, one has to assume here, that \mathcal{O} is a *free* \mathfrak{S} -algebra, generated by \mathfrak{C} .

Therefore we need the notion of a *partial* \mathfrak{S} -algebra, where \mathcal{R} is defined not on all of $\mathfrak{S}(\mathcal{M})$, but on a subcategory $\mathfrak{R}(\mathcal{M}) \subseteq \mathfrak{S}(\mathcal{M})$.

Since \mathfrak{S} is a 2-monad, we need, in addition, to choose subcategories $\mathfrak{R}^k(\mathcal{M}) \subseteq \mathfrak{S}^k(\mathcal{M})$ for all $k \geq 1$, s.t. iterating \mathcal{R} , and applying \mathcal{K}^* does not take us out of this sequence of subcategories. This is the contents of the following definition.

Definition 23 A (partial) \mathfrak{S} -algebra is $(\mathcal{M}, \mathcal{R}, \{\mathfrak{R}^k(\mathcal{M})\}_{k \geq 1})$, where \mathcal{M} is a category, $\mathfrak{R}^k(\mathcal{M}) \subseteq \mathfrak{S}^k(\mathcal{M})$ are subcategories, and $\mathcal{R} : \mathfrak{R}(\mathcal{M}) \rightarrow \mathcal{M}$ is a functor, s.t.

1. For any $k \geq 0, n > k$ we have

$$\mathfrak{S}^k(\mathcal{R})(\mathfrak{R}^n(\mathcal{M})) \subseteq \mathfrak{R}^{n-k-1}(\mathcal{M}), \quad (43)$$

where we put $\mathfrak{R}^0(\mathcal{M}) := \mathcal{M}$. For any $k \geq 0, n > k + 1$ we have

$$\mathfrak{S}^k(\mathcal{K}^*)(\mathfrak{R}^n(\mathcal{M})) \subseteq \mathfrak{R}^{n-k-1}(\mathcal{M}). \quad (44)$$

For any $k \geq 0, n \geq 0$ we have

$$\mathfrak{S}^k(\epsilon)(\mathfrak{R}^n(\mathcal{M})) \subseteq \mathfrak{R}^{n+k+1}(\mathcal{M}). \quad (45)$$

2. The following diagrams are commutative up to coherent natural isomorphisms

$$\begin{array}{ccc} \mathfrak{R}^2(\mathcal{M}) & \xrightarrow{\mathfrak{S}(\mathcal{R})} & \mathfrak{R}(\mathcal{M}) \\ \mathcal{K}^* \downarrow & & \downarrow \mathcal{R} \\ \mathfrak{R}(\mathcal{M}) & \xrightarrow{\mathcal{R}} & \mathcal{M} \end{array} \quad \begin{array}{ccc} \mathcal{M} & \xrightarrow{\epsilon} & \mathfrak{R}(\mathcal{M}) \\ & \searrow & \downarrow \mathcal{R} \\ & & \mathcal{M} \end{array} \quad (46)$$

Conditions (43), (44), (45) require that the sequence $\{\mathfrak{R}^k(\mathcal{M})\}$ is closed with respect to the structure maps, given by \mathfrak{S} . Without them one cannot define coherence in part 2.

Let \mathcal{M}, \mathcal{N} be two partial \mathfrak{S} -algebras. A functor $\mathcal{F} : \mathcal{M} \rightarrow \mathcal{N}$ is a **morphism of partial \mathfrak{S} -algebras**, if for any $k \geq 1$ we have

$$\mathfrak{S}^k(\mathcal{F})(\mathfrak{R}^k(\mathcal{M})) \subseteq \mathfrak{R}^k(\mathcal{N}), \quad (47)$$

and the diagram

$$\begin{array}{ccc} \mathfrak{R}(\mathcal{M}) & \xrightarrow{\mathcal{R}} & \mathcal{M} \\ \mathfrak{S}(\mathcal{F}) \downarrow & & \downarrow \mathcal{F} \\ \mathfrak{R}(\mathcal{N}) & \xrightarrow{\mathcal{R}} & \mathcal{N} \end{array} \quad (48)$$

is commutative (up to coherent natural isomorphisms).

A double category $(\mathcal{O}, \mathcal{M})$ is a **partial \mathfrak{S} -category**, if both \mathcal{O}, \mathcal{M} are partial \mathfrak{S} -algebras, and all the structure functors are morphisms of partial \mathfrak{S} -algebras. It is clear how to extend the notion of generators of an \mathfrak{S} -category to partial \mathfrak{S} -categories.

3.3 Simplicial diagrams of nested graphs

Here we define a partial \mathfrak{S} -category structure on $\mathfrak{N}^{op} := (\mathfrak{D}^{op}, \mathfrak{M}^{op})$. Taking the opposite categorical structure is necessary, because we define the \mathfrak{S} -algebra structures on $\mathfrak{D}^{op}, \mathfrak{M}^{op}$ by considering them as subcategories of Cat , and hence we have to reverse dependencies back into the directions of functors.

First we explain why \mathfrak{N}^{op} has only a partial \mathfrak{S} -structure, and what is the nature of restrictions \mathfrak{R}^k that we make.

We would like to see $\mathcal{R}(\mathcal{D}_S)$, for $\mathcal{D}_S \in \mathfrak{S}(\mathfrak{D}^{op})$, as the result of gluing of graphs in the image of \mathcal{D}_S . The images in $\mathcal{R}(\mathcal{D}_S)$ of the components of \mathcal{D}_S should be full subgraphs of $\mathcal{R}(\mathcal{D}_S)$.

Here we are led to the following question: when does a morphism of diagrams $\mathcal{D}_S \rightarrow \mathcal{D}_T$ correspond to $\mathcal{R}(\mathcal{D}_S) \rightarrow \mathcal{R}(\mathcal{D}_T)$ being a full subgraph? It is only the morphisms that have this property, that should be allowed in $\mathfrak{R}(\mathfrak{D}^{op})$, and similarly for $\mathfrak{R}(\mathfrak{M}^{op})$.

The answer includes requiring that each component of $\mathcal{D}_S \rightarrow \mathcal{D}_T$ is a dependency, but this is not enough. One should also require the morphism between diagrams to be a *fibration*. We define this notion here, using the usual notion of fibrations between simplicial sets. We formulate our definitions in the context of a general category \mathcal{M} , so that we can apply the resulting machinery to $\mathfrak{D}^{op}, \mathfrak{M}^{op}$, and in general to any category of labeled graphs.

Let \mathcal{M} be a category, having a set \mathcal{I} of generators.¹² Our main example for $(\mathcal{M}, \mathcal{I})$ is $(\mathfrak{D}, \mathfrak{C})$, where \mathfrak{C} is the set of corollas in \mathfrak{D} . Let $\mathcal{D}_S : \mathcal{S} \rightarrow \mathcal{M}$ be an object in $\mathfrak{S}(\mathcal{M})$.

Definition 24 *Let $I \in \mathcal{I}$. An I -point in \mathcal{D}_S , is a morphism $I[n] \rightarrow \mathcal{D}_S$ in $\mathfrak{S}(\mathcal{M})$, where $I[n]$ is I , considered as a constant diagram over $\Delta[n]$.*¹³

It is clear that for each such $I \in \mathcal{I}$ we obtain a simplicial set $I(\mathcal{D}_S)$ of I -points. If $(F, \phi) : \mathcal{D}_S \rightarrow \mathcal{D}_T$ is a morphism in $\mathfrak{S}(\mathcal{M})$, it induces a morphism of simplicial sets $(F, \phi)_I : I(\mathcal{D}_S) \rightarrow I(\mathcal{D}_T)$.

¹²Here we use the term “generator” in the usual categorical meaning.

¹³Here $\Delta[n]$ is the standard n -simplex.

Definition 25 We will say that (F, ϕ) is a **fibration**, if for each $I \in \mathcal{I}$ the morphism of simplicial sets $(F, \phi)_I$ is a fibration.

We will call (F, ϕ) a **dependency**, if F is injective, and (F, ϕ) is a fibration. We will denote by $\mathfrak{D}(\mathcal{M})$ the subcategory of $\mathfrak{S}(\mathcal{M})$, consisting of dependencies.

It is clear that our definitions depend not only on the category \mathcal{M} , but also on the choice of generators \mathcal{I} . Since $\{\Delta[n]\}_{n \geq 0}$ is a set of generators for $SSet$, in many important examples, e.g. $\mathfrak{D}^{op}, \mathfrak{M}^{op}$, it is easy to see that $\mathfrak{D}(\mathcal{I}) := \{I[n]\}_{n \geq 0; I \in \mathcal{I}}$ is a set of generators for $\mathfrak{D}(\mathcal{M})$. If we can, we will always assume that for an \mathcal{M} under consideration, $\mathfrak{D}(\mathcal{I})$ is chosen as the set of generators for $\mathfrak{D}(\mathcal{M})$.

One could extend \mathfrak{D} to a monad on pairs $(\mathcal{M}, \mathcal{I})$, where \mathcal{M} is a category and \mathcal{I} is a set of generators. Functors between such pairs should map generators to generators. However, this is too narrow for our purposes, since very often it will happen that functors that we consider do not preserve generators, e.g. the source functor $\mathfrak{M} \rightarrow \mathfrak{D}$, or the action of \mathfrak{S} on \mathfrak{D} .

Now we are ready to define a partial \mathfrak{S} -category structure on $(\mathfrak{D}^{op}, \mathfrak{M}^{op})$, using \mathfrak{D} introduced above (Definition 25). The first step is choosing generators for $\mathfrak{D}^{op}, \mathfrak{M}^{op}$. The following obvious proposition provides them (here again, by generators we mean the usual categorical ones).

Proposition 18 Let $\mathfrak{D}, \mathfrak{M}$ be the categories of objects and morphisms in the category of nested graphs \mathfrak{N} . Let $\mathfrak{C} \subset \text{Obj}(\mathfrak{D})$ consist of corollas, and let $\mathfrak{M}\mathfrak{C} \subset \text{Obj}(\mathfrak{M})$ consist of those morphisms, that have corollas as codomains. Then $\mathfrak{C}, \mathfrak{M}\mathfrak{C}$ are generators for $\mathfrak{D}^{op}, \mathfrak{M}^{op}$ respectively.

It is easy to see that $\mathfrak{D}(\mathfrak{C}), \mathfrak{D}(\mathfrak{M}\mathfrak{C})$ are generators for $\mathfrak{D}(\mathfrak{D}^{op}), \mathfrak{D}(\mathfrak{M}^{op})$, and so on. Therefore we have sequences of categories

$$\mathfrak{D}^k(\mathfrak{D}^{op}) \subset \mathfrak{S}^k(\mathfrak{D}^{op}), \quad \mathfrak{D}^k(\mathfrak{M}^{op}) \subset \mathfrak{S}^k(\mathfrak{M}^{op}), \quad \forall k \geq 1. \quad (49)$$

These sequences satisfy the following compatibility conditions.

Proposition 19 Let $\iota : \mathfrak{D}^{op} \rightarrow \mathfrak{M}^{op}$, $\sigma, \tau : \mathfrak{M}^{op} \rightarrow \mathfrak{D}^{op}$ be the identity, source, and target functors respectively. Then for any $k \geq 1$

$$\mathfrak{S}^k(\iota)(\mathfrak{D}^k(\mathfrak{D}^{op})) \subseteq \mathfrak{D}^k(\mathfrak{M}^{op}), \quad \mathfrak{S}^k(\sigma), \mathfrak{S}^k(\tau)(\mathfrak{D}^k(\mathfrak{M}^{op})) \subseteq \mathfrak{D}^k(\mathfrak{D}^{op}). \quad (50)$$

Proof: Since $\tau : \mathfrak{M}^{op} \rightarrow \mathfrak{D}^{op}$ maps generators to generators, it is clear that it satisfies (50). It is straightforward to prove the same statement about ι .

To prove that $\mathfrak{S}^k(\sigma)$ preserves dependencies, we use the fact that $\mathfrak{S}^k(\tau)$ does it. We give details only for $k = 1$, the rest is similar.

Let $\bar{\phi} : (\bar{\mu}, \bar{\kappa}) \rightarrow (\bar{\mu}', \bar{\kappa}')$ be a morphism in $\mathfrak{D}(\mathfrak{M}^{op})$, where $(\bar{\mu}, \bar{\kappa}) : \bar{\mathcal{G}} \rightarrow \bar{\mathcal{H}}$ is a diagram over \mathcal{S} , $(\bar{\mu}', \bar{\kappa}') : \bar{\mathcal{G}}' \rightarrow \bar{\mathcal{H}}'$ is a diagram over \mathcal{T} ; \mathcal{S}, \mathcal{T} being finite simplicial sets.

By definition of \mathfrak{M} , $\bar{\phi}$ is a triple $(F, \bar{\delta}, \bar{\delta}')$, where $F : \mathcal{S} \rightarrow \mathcal{T}$ is an injective map of simplicial sets, and $\bar{\delta} : \bar{\mathcal{G}} \rightarrow \bar{\mathcal{G}}'$, $\bar{\delta}' : \bar{\mathcal{H}} \rightarrow \bar{\mathcal{H}}'$ are fibrations.

To prove that σ satisfies (50) we should show that for any node A in $\bar{\mathcal{G}}'$, if A is in the image of $\bar{\delta}$, then so is any pre-image B of A in $\bar{\mathcal{G}}'$. The analogous statement is obviously true for images of A, B in $\bar{\mathcal{H}}'$, and therefore B is in the image of $\bar{\delta}$ because, by definition of morphisms in \mathfrak{M} , image of $\bar{\delta}$ is the pre-image of the image of $\bar{\delta}'$. ■

Now we define the partial \mathfrak{S} -algebra structures on $\mathfrak{D}^{op}, \mathfrak{M}^{op}$. We use the fact, proved in Proposition 16, that any finitely cocomplete category is automatically an \mathfrak{S} -algebra, with the action given by taking colimits of simplicial diagrams.

In our case the finitely cocomplete category is the category of small categories Cat . We start with objects in the category of objects \mathfrak{D}^{op} . Let $\bar{\mathcal{G}} \in \mathfrak{D}(\mathfrak{D}^{op})$, we define

$$\mathcal{R}(\bar{\mathcal{G}}) := \text{colim}(\bar{\mathcal{G}}), \quad (51)$$

with the colimit taken in Cat .

Proposition 20 *Defined as above, $\mathcal{R}(\bar{\mathcal{G}})$ is an object in \mathfrak{D}^{op} .*

Proof: It is obvious that $\mathcal{R}(\bar{\mathcal{G}})$ is a finite category. We claim it is also direct, i.e. it is a nested graph. Indeed, it is clear that if $\delta : \mathcal{G}_2 \rightarrow \mathcal{G}_1$ is a dependency between two graphs, then it preserves the gradings on $\mathcal{G}_1, \mathcal{G}_2$, given by height (Definition 9). Therefore this grading is invariant with respect to morphisms in the diagram $\bar{\mathcal{G}}$, and hence it induces a grading on the colimit, i.e. the colimit is a direct category. ■

Next we would like to show that \mathcal{R} from (51) can be extended to a map $Obj(\mathfrak{D}(\mathfrak{M})) \rightarrow Obj(\mathfrak{M})$. First we prove that colimit of a diagram of direct functors is a direct functor.

Proposition 21 *Let $\mathcal{D}_{\mathcal{S}} : \mathcal{S} \rightarrow \mathfrak{M}^{op}$ be an object in $\mathfrak{D}(\mathfrak{M}^{op})$. Let $\bar{\phi} : \bar{\mathcal{G}} \rightarrow \bar{\mathcal{G}}'$ be $\mathcal{D}_{\mathcal{S}}$, considered as a diagram of direct functors between graphs (i.e. we forget the factorizations into mergers/contractions). Then $\phi := \text{colim}(\bar{\phi})$ is a direct functor between graphs.*

Proof: From Proposition 20 we know that $\mathcal{G} := \text{colim}(\overline{\mathcal{G}}), \mathcal{G}' := \text{colim}(\overline{\mathcal{G}}')$ are nested graphs. So all we have to prove is that $\text{colim}(\overline{\phi})$ is a direct functor.

First we make the following important observation: let A be a node in \mathcal{G} , if A is a ϕ -vertex, then for any simplex $s \in \mathcal{S}$, and any pre-image A_s of A with respect to the canonical $\mathcal{G}_s \rightarrow \mathcal{G}$, the node A_s is a ϕ_s -vertex, and conversely, if A is not a ϕ -vertex, then A_s is not a ϕ_s -vertex *for any* $s \in \mathcal{S}$.

Suppose that A is a ϕ -vertex, and let $A_s \in \mathcal{G}_s$ be a non- ϕ_s -vertex, that is mapped to A . Let f_s be a flag, decorated by A_s , and contracted by ϕ_s . It is clear that $\mathcal{G}_s \rightarrow \mathcal{G}$ does not contract flags (to see this consider the grading by height, as in the proof of Proposition 20), and hence f_s has a non-trivial image f in \mathcal{G} , which has to be contracted by ϕ , since also $\mathcal{G}'_s \rightarrow \mathcal{G}'$ does not contract flags. Clearly f is decorated by A – contradiction.

Suppose now that A is not a ϕ -vertex, and let A_s be a pre-image of A in \mathcal{G}_s . Let f be a flag in \mathcal{G} , decorated by A and contracted by ϕ , let $f_t \in \mathcal{G}_t$ be a pre-image of f . Then clearly f_t is contracted by ϕ_t . Suppose there is a simplex $u \in \mathcal{S}$ and a node $A_u \in \mathcal{G}_u$, s.t. one boundary of A_u is A_s and another is A_t . According to Proposition 3 there is a flag f_u , decorated by A_u , and contracted by ϕ_u . Consequently A_s cannot be a ϕ_s vertex either.

It is easy to see that if there is no such A_u , gluing A_s and A_t , there has to be a sequence of A_u 's, gluing vertices, s.t. at the end A_s and A_t are glued (otherwise they cannot have the same image in \mathcal{G}). Applying the above analysis to each one of these A_u 's we conclude that A_s is not a ϕ_s -vertex.

Now let $A \in \mathcal{G}$ be a node, that is contracted by ϕ , and let f be a flag, decorated by A , that is not contracted by ϕ . Clearly, there is $s \in \mathcal{S}$, and a flag $f_s \in \mathcal{G}_s$, that is mapped to f by $\mathcal{G}_s \rightarrow \mathcal{G}$. Let A_s be the decoration of f_s . According to the observation above, A_s is not a ϕ_s -vertex, and hence there is a ϕ_s vertex $B_s \in \mathcal{G}_s$ and a flag g_s , decorated by B_s , s.t. $\phi_s(B_s) = \phi_s(A_s)$, and $\phi_s(g_s)$ factors $\phi_s(f_s)$.

According to the observation above, the image B of B_s in \mathcal{G} is a ϕ -vertex, and it is obvious that $\phi(g)$ factors $\phi(f)$, where g is the image of g_s . ■

Now let \mathcal{D} be an object in $\mathfrak{D}(\mathfrak{M}^{op})$. Since, by definition, every object in \mathfrak{M}^{op} is a merger, followed by a contraction, we can decompose \mathcal{D} into $\overline{\mu}$, $\overline{\kappa}$, where the former is a diagram of mergers, and the latter is a diagram of contractions.

According to Proposition 21 the functors $\mu := \text{colim}(\overline{\mu}), \kappa := \text{colim}(\overline{\kappa})$ are direct functors. It is not true, in general, that μ is a merger, but κ has to be a contraction, as the following proposition shows.

Proposition 22 *Let $\bar{\phi}$ be as in Proposition 21. If $\bar{\phi}$ is a diagram of contractions, then $\phi := \text{colim}(\bar{\phi})$ is a contraction as well.*

Proof: We have to show that for any node $A \in \mathcal{G}'$, the fiber $\phi^{-1}(A)$ is a corolla in \mathcal{G} . Suppose it is not, then let B, B' be two distinct vertices in $\phi^{-1}(A)$, and let B_s, B'_t be pre-images of B, B' in $\mathcal{G}_s, \mathcal{G}_t$ respectively.

According to the observation, we made in the proof of Proposition 21, B_s, B'_t have to be ϕ_s, ϕ_t -vertices, and therefore $\phi_s(B_s) \neq \phi_t(B'_t)$ (since fibers of contractions are corollas). On the other hand the images of $\phi_s(B_s), \phi_t(B'_t)$ in \mathcal{G}' are equal, i.e. $\phi_s(B_s), \phi_t(B'_t)$ are glued together in the diagram $\bar{\mathcal{G}}'$.

If there is a simplex $u \in \mathcal{S}$, and a node $X_u \in \mathcal{G}'_u$, s.t. $\phi_s(B_s), \phi_t(B'_t)$ are its boundaries, then taking the pre-image of X_u in \mathcal{G}_u we find that B_s, B'_t are glued as well. If there is no such X_u , there has to be a sequence of gluings, resulting in identification of $\phi_s(B_s), \phi_t(B'_t)$ in \mathcal{G}' . Taking pre-images at each step of the sequence we obtain that still $B = B'$. ■

Now having $\mathcal{D} \in \mathfrak{D}(\mathfrak{M}^{op})$, and its decomposition into $\bar{\mu}, \bar{\kappa}$, we define

$$\mathcal{R}(\mathcal{D}) := (\mu', \kappa \circ \kappa') \in \mathfrak{M}^{op}, \quad (52)$$

where $\kappa := \text{colim}(\bar{\kappa})$, and (μ', κ') is the factorization of $\mu := \text{colim}(\bar{\mu})$ into a merger, followed by a contraction, given by (13).

The following statement is the reason we have introduced $\mathfrak{D} \subset \mathfrak{S}$. In general, it is not true that taking colimit of a morphism in $\mathfrak{S}(\mathfrak{D}^{op})$ we obtain a morphism in \mathfrak{D}^{op} , but if the start with a morphism in $\mathfrak{D}(\mathfrak{D}^{op})$, it is true.

Proposition 23 *The maps from (51), (52) extend to functors*

$$\mathcal{R} : \mathfrak{D}(\mathfrak{D}^{op}) \rightarrow \mathfrak{D}^{op}, \quad \mathcal{R} : \mathfrak{D}(\mathfrak{M}^{op}) \rightarrow \mathfrak{M}^{op}, \quad (53)$$

and these functors commute with $\iota : \mathfrak{D}^{op} \rightarrow \mathfrak{M}^{op}$, $\sigma, \tau : \mathfrak{M}^{op} \rightarrow \mathfrak{D}^{op}$.

Proof: Let $\bar{\mathcal{G}}, \bar{\mathcal{H}}$ be two objects in $\mathfrak{D}(\mathfrak{D})$, and let $\bar{\delta} : \bar{\mathcal{G}} > \bar{\mathcal{H}}$ be a dependency. Taking colimit in \mathcal{Cat} we obtain two graphs \mathcal{G}, \mathcal{H} , and a functor $\delta : \mathcal{H} \rightarrow \mathcal{G}$. We claim that δ is injective, and its image is a full subgraph.

To see that δ is injective one should look at the way \mathcal{G} is built. It is the colimit of the diagram $\bar{\mathcal{G}}$, and every morphism in this diagram is a dependency. Consequently \mathcal{G} is obtained by gluing objects in $\bar{\mathcal{G}}$ along morphisms.

In addition every component of $\bar{\delta}$ is a dependency, and for any node A in the image of $\bar{\delta}$, any pre-image of A is also in the image of $\bar{\delta}$. This means that if two nodes or flags in the image of $\bar{\delta}$ are glued together in $\bar{\mathcal{G}}$, then the nodes/flags that perform the gluing, are themselves in the image of $\bar{\delta}$.

Now it is clear that injectivity of δ follows from injectivity of components of $\bar{\delta}$. Similarly the fact that images of components of $\bar{\delta}$ are full subgraphs implies that δ also has a full subgraph as the image.

Once we know that \mathcal{R} extends to morphisms in \mathfrak{D}^{op} by taking colimits, its functoriality follows from functoriality of colimits.

Applied to diagrams of pairs of dependencies, that are morphisms in $\mathfrak{D}(\mathfrak{M}^{op})$, we obtain pairs of dependencies, and it is easy to see, that these pairs are dependencies between morphisms (i.e. the condition on images/pre-images is satisfied). Again, functoriality follows from functoriality of colimits.

Finally, commutativity of \mathcal{R} with the identity, source, and target functors is obvious, since we define \mathcal{R} on $\mathfrak{D}(\mathfrak{M}^{op})$ by extending its definition on $\mathfrak{D}(\mathfrak{D}^{op})$. ■

Now we can formulate the main proposition of this section.

Proposition 24 *The functors from (53) make $\mathfrak{N}^{op} = (\mathfrak{D}^{op}, \mathfrak{M}^{op})$ into a partial \mathfrak{S} -category, i.e. $(\mathfrak{D}^{op}, \mathcal{R}, \{\mathfrak{D}^k(\mathfrak{D}^{op})\})$, $(\mathfrak{M}^{op}, \mathcal{R}, \{\mathfrak{D}^k(\mathfrak{M}^{op})\})$ are partial \mathfrak{S} -algebras, the functors ι, σ, τ are morphisms of \mathfrak{S} -algebras, and the following diagram is well defined and commutative*

$$\begin{array}{ccc} \mathfrak{D}(\mathfrak{M})_{\mathfrak{D}(\sigma)} \coprod_{\mathfrak{D}(\tau)} \mathfrak{D}(\mathfrak{M}) & \xrightarrow{\mathfrak{S}(\gamma)} & \mathfrak{D}(\mathfrak{M}) \\ \mathcal{R} \coprod \mathcal{R} \downarrow & & \downarrow \mathcal{R} \\ \mathfrak{M}_{\sigma} \coprod_{\tau} \mathfrak{M} & \xrightarrow{\gamma} & \mathfrak{M} \end{array} \quad (54)$$

Proof: Since we know that $(Cat, colim)$ is an \mathfrak{S} -algebra (Proposition 16), all we have to prove here, to show that $\mathfrak{D}^{op}, \mathfrak{M}^{op}$ are \mathfrak{D} -algebras, is that taking colimits of the diagrams that we consider, we get objects and morphisms in the right categories, i.e. we have to show that equations (43), (44) are satisfied for both \mathfrak{D}^{op} and \mathfrak{M}^{op} . Validity of (45) is obvious.

1. For any $k \geq 0, n > k + 1$ $\mathfrak{S}^k(\mathcal{K}^*)(\mathfrak{D}^n(\mathfrak{D}^{op})) \subseteq \mathfrak{D}^{n-k-1}(\mathfrak{D}^{op})$ and similarly for \mathfrak{M}^{op} . This is almost obvious: \mathcal{K}^* is defined by taking direct sums of simplicial sets, and then taking the diagonal in a bisimplicial set. Both operations preserve injectivity and map fibrations to fibrations. When we apply \mathfrak{S} to \mathcal{K}^* , everything is the same, just happens component-wise.

2. For any $k \geq 0, n > k$ we have $\mathfrak{S}^k(\mathcal{R})(\mathfrak{D}^n(\mathfrak{D}^{op})) \subseteq \mathfrak{N}^{n-k-1}(\mathfrak{D}^{op})$, and similarly for \mathfrak{M}^{op} . This is less obvious than the previous property, but the proof is straightforward.

To finish the proof we have to show that (54) is commutative. If we did not care for factorization of direct functors into mergers/contractions, commutativity is clear: it just follows from functoriality of taking colimit.

So all we have to do is to compare the factorizations, obtained in two paths of (54). To do this, it is enough to compare the mergers, and to compare the mergers it is enough to compare the nodes that get merged.

It is clear that $\mathcal{R}(\mathfrak{S}(\gamma)(\overline{\kappa}, \overline{\kappa}')) = \gamma(\mathcal{R}(\overline{\kappa}), \mathcal{R}(\overline{\kappa}'))$, because there are no mergers performed, only contractions. Also $\mathcal{R}(\mathfrak{S}(\gamma)(\overline{\mu}, \overline{\mu}')) = \gamma(\mathcal{R}(\overline{\mu}), \mathcal{R}(\overline{\mu}'))$ is clear. Here all nodes are vertices in fibers, and on both sides factorizations into mergers/contractions are unique.

The only non-trivial case is $\mathcal{R}(\mathfrak{S}(\gamma)(\overline{\kappa}, \overline{\mu})) = \gamma(\mathcal{R}(\overline{\kappa}), \mathcal{R}(\overline{\mu}))$. This is true, because the only nodes that are merged are the vertices in fibers, and we know that as far as fibers are concerned \mathcal{R} maps vertices to vertices, and non-vertices to non-vertices (proof of Proposition 21). ■

3.4 Graph-algebras

In this section we define the central notion of a *graph-algebra*. This is the generalization to the nested context of abstract categories of labeled plain graphs (section 3.1). We could have used the term “abstract categories of labeled nested graphs”, but it is rather long, and we have opted for “graph-algebra” instead.

As in the case of plain graphs, graph-algebras are given by axiomatization of the important properties of \mathfrak{N} . We already know that $\mathfrak{N}^{op} = (\mathfrak{D}^{op}, \mathfrak{M}^{op})$ is a partial \mathfrak{S} -category (section 3.3). The other important property of \mathfrak{N}^{op} is given in the following proposition.

Proposition 25 *The partial \mathfrak{S} -category $(\mathfrak{D}^{op}, \mathfrak{M}^{op})$ is generated (as an \mathfrak{S} -category, section 3.2) by the set of corollas $\mathfrak{C} \subset \text{Obj}(\mathfrak{D})$.*

Proof: Let $\mathcal{G} \in \mathfrak{D}$ be a nested graph. We define a simplicial set $\mathcal{S}(\mathcal{G})$ as follows: $\mathcal{S}(\mathcal{G})_0$ is the set of components of \mathcal{G} (Definition 7), if we have defined $\mathcal{S}(\mathcal{G})_k$, then $\mathcal{S}(\mathcal{G})_{k+1}$ is the set of components in pairwise intersections of the corollas, corresponding to elements of $\mathcal{S}(\mathcal{G})_k$.¹⁴ The boundary and degeneration maps are obvious, and so we have a diagram

$$\mathcal{D}_{\mathcal{S}(\mathcal{G})} : \mathcal{S}(\mathcal{G}) \rightarrow \mathfrak{D}^{op}. \quad (55)$$

¹⁴Note that we do not require the elements in the pairs to be distinct, and hence at level $k + 1$ we have copies of graphs from level k

Clearly this diagram is an object in $\mathfrak{D}(\mathfrak{D}^{op})$ (since inclusions of components are dependencies), and clearly all graphs in this diagram are corollas. Taking \mathcal{R} we get \mathcal{G} back.

Similarly, given an object $(\bar{\mu}, \bar{\kappa})$ in \mathfrak{M} , we can decompose it into generators by taking components in the codomain, their inverse images in the domain, and restricting the merger/contraction to them. Iterating as above, we get a diagram $\mathcal{D}_{S(\bar{\mu}, \bar{\kappa})} \in \mathfrak{D}(\mathfrak{M}^{op})$, consisting of generators. Clearly $\mathcal{R}(\mathcal{D}_{S(\bar{\mu}, \bar{\kappa})}) = (\bar{\mu}, \bar{\kappa})$.

If $\mathcal{H} < \mathcal{G}$ is a morphism in \mathfrak{D}^{op} , we can obtain it as an \mathfrak{S} -product of morphisms, having corollas as codomains, by decomposing \mathcal{G} into components, taking inverse images in \mathcal{H} ,¹⁵ taking intersections as above, and iterating. Similarly for morphisms in \mathfrak{M}^{op} . ■

In section 3.1 we have defined abstract categories of labeled plain graphs as monoidal categories, having a faithful monoidal functor to \mathfrak{P} , and being generated by corollas. The functor to \mathfrak{P} is the bookkeeping device, that allows us to track combinatorics of compositions of operations in labeled graphs.

We would like to do the same in the case of nested graphs, and use \mathfrak{N} as the bookkeeping device, however, for many purposes, in particular for defining higher categories, \mathfrak{N} is not enough to keep books on all operations.

An example of operation, that cannot be described by \mathfrak{N} is association of the identity automorphism to an object. This is basic for higher categories, yet one cannot parameterize it by a merger/contraction between nested graphs.

If a category has only one object, then the identity automorphism is just the unit in the monoid of endomorphisms of the object. If a category has more than one object, one has to choose a unit for each one of the monoids.

In the language of plain operads, objects like units in monoids are described as constants (i.e. 0-ary operations) in the corresponding operads. In the nested context, 0-ary operations have to be given as *functions* of other operations, as for example identity automorphisms are functions of objects.

The following definition provides the systematic way to define constants in nested operads.

Definition 26 An \mathfrak{S} -category of nested graphs with constants is a partial \mathfrak{S} -category $\mathfrak{N}_c^{op} = (\mathfrak{D}_c^{op}, \mathfrak{M}_c^{op})$, and a morphism of partial \mathfrak{S} -categories $\mathfrak{K} : \mathfrak{N}^{op} \rightarrow \mathfrak{N}_c^{op}$, such that $\mathfrak{K}_{\mathfrak{D}} : \mathfrak{D}^{op} \rightarrow \mathfrak{D}_c^{op}$, $\mathfrak{K}_{\mathfrak{M}} : \mathfrak{M}^{op} \rightarrow \mathfrak{M}_c^{op}$ are faithful, and \mathfrak{N}_c^{op} is generated (as a partial \mathfrak{S} -category) by $\mathfrak{K}_{\mathfrak{D}}(\mathfrak{C})$.

¹⁵Some inverse images might be empty, but we allow empty graphs.

In short, \mathfrak{N}_c is obtained from \mathfrak{N} by adding some objects to \mathfrak{M} , that have corollas as targets, and then closing with respect to γ and the \mathfrak{S} -action.

There are some special conditions, one might require (and we will usually do), that the added objects in \mathfrak{M} satisfy. One can require that constants are right inverses of dependencies in the following sense: let $\delta : \mathcal{G} > \mathcal{H}$ be a morphism in \mathfrak{D} . We will say that an object $\phi \in \mathfrak{M}_c$ is a **right inverse** of δ if $\sigma(\phi) = \mathcal{H}$, $\tau(\phi) = \mathcal{G}$, and there is a morphism $(Id_{\mathcal{H}}, \delta) : \phi > Id_{\mathcal{H}}$ in \mathfrak{M}_c . A simple example of this definition is the following morphism

$$\cdot \longrightarrow \cdot \rightarrow \cdot \quad (56)$$

which we suppose to be a right inverse to both of the boundaries in the opposite direction. This is the morphism representing association of the identity automorphism to an object.

Let $\mathfrak{N}_c, \mathfrak{N}_{c'}$ be two categories of nested graphs with constants. A **morphism** from \mathfrak{N}_c to $\mathfrak{N}_{c'}$ is a morphism $\mathfrak{N}_c^{op} \rightarrow \mathfrak{N}_{c'}^{op}$ of partial \mathfrak{S} -categories under \mathfrak{N}^{op} , i.e. the following diagram is commutative

$$\begin{array}{ccc} \mathfrak{N}_c^{op} & \xrightarrow{\quad} & \mathfrak{N}_{c'}^{op} \\ & \nwarrow \mathfrak{K} \quad \nearrow \mathfrak{K}' & \\ & \mathfrak{N}^{op} & \end{array} \quad (57)$$

Definition 27 A **graph-algebra** is a pair $(\mathfrak{G}, \mathfrak{U})$, where $\mathfrak{G} = (\mathfrak{D}_{\mathfrak{G}}, \mathfrak{M}_{\mathfrak{G}})$ is a double category, having a partial \mathfrak{S} -category structure on $(\mathfrak{D}_{\mathfrak{G}}^{op}, \mathfrak{M}_{\mathfrak{G}}^{op})$, and \mathfrak{U} is a morphism of partial \mathfrak{S} -categories $\mathfrak{U} : \mathfrak{G}^{op} \rightarrow \mathfrak{N}_c^{op}$, for some \mathfrak{N}_c as in Definition 26, s.t. $(\mathfrak{D}_{\mathfrak{G}}^{op}, \mathfrak{M}_{\mathfrak{G}}^{op})$ is generated (as a partial \mathfrak{S} -category) by $\mathfrak{U}^{-1}(\mathfrak{C}) \subseteq \text{Obj}(\mathfrak{D}_{\mathfrak{G}})$.

A **morphism between graph-algebras** $(\mathfrak{G}, \mathfrak{U}), (\mathfrak{G}', \mathfrak{U}')$ is a pair $(\mathfrak{F}, \mathfrak{F}_{\mathfrak{N}})$ of morphisms of partial \mathfrak{S} -categories, s.t. the following diagram is commutative

$$\begin{array}{ccc} \mathfrak{G}^{op} & \xrightarrow{\quad \mathfrak{F} \quad} & \mathfrak{G}'^{op} \\ \mathfrak{U} \downarrow & & \downarrow \mathfrak{U}' \\ \mathfrak{N}_c^{op} & \xrightarrow{\quad \mathfrak{F}_{\mathfrak{N}} \quad} & \mathfrak{N}_{c'}^{op} \end{array} \quad (58)$$

and $\mathfrak{F}_{\mathfrak{N}}$ is a morphism of categories of nested graphs with constants.

We will usually denote $(\mathfrak{G}, \mathfrak{U})$ simply by \mathfrak{G} , and $(\mathfrak{F}, \mathfrak{F}_{\mathfrak{N}})$ by \mathfrak{F} , if it does not lead to confusion.

Recall, that having an \mathfrak{S} -algebra \mathcal{M} , we have an \mathfrak{S} -category of morphisms $(\mathcal{M}, \text{Mor}(\mathcal{M}))$.

Definition 28 Let \mathfrak{G} be a graph-algebra. A \mathfrak{G} -operad in an \mathfrak{S} -algebra \mathcal{M}^{op} is a morphism of partial \mathfrak{S} -categories

$$\mathfrak{R} : \mathfrak{G}^{op} \rightarrow (\mathcal{M}^{op}, Mor(\mathcal{M}^{op})). \quad (59)$$

Note that, contrary to the case of abstract categories of labeled plain graphs, we do not require that \mathfrak{R} maps a particular class of morphisms to isomorphisms.

The reason is clear: in \mathfrak{P} the monoidal structure did not provide all the information on decomposition of graphs into corollas, and we needed atomization morphisms. In \mathfrak{N} , and in graph-algebras in general, all the information on breaking graphs into subgraphs is contained in the \mathfrak{S} -action, and all we need to require is \mathfrak{S} -equivariance of \mathfrak{R} .

Nested operads, introduced in Definition 28, are algebraic objects, tailored to describe operations, that have a hierarchy of interdependencies. It is natural, therefore, to ask the following question: let $\{(\mathfrak{G}_i, \mathfrak{R}_i)\}_{i \in \mathcal{I}}$ be a set of nested operads in a given \mathfrak{S} -algebra \mathcal{M} , what is the nested operad $(\overline{\mathfrak{G}}, \overline{\mathfrak{R}})$, parameterizing operations between these operads?

We could have formulated this question differently, asking for a category, having $\{(\mathfrak{G}_i, \mathfrak{R}_i)\}_{i \in \mathcal{I}}$ as objects, but since the term “higher category” is already widely used, and, more importantly, $\{(\mathfrak{G}_i, \mathfrak{R}_i)\}_{i \in \mathcal{I}}$ do not have to be only objects in $(\overline{\mathfrak{G}}, \overline{\mathfrak{R}})$, we prefer to say operad of operations between operads.

Just as a \mathfrak{G} -operad depends on the choice of \mathfrak{G} , so does $\overline{\mathfrak{G}}$ depend on the choice of the ambient graph-algebra, where \mathfrak{G}_i ’s live. If we take commutative algebras, for example, they constitute a category on their own, or part of the category of associative algebras and so on, i.e. given a commutative algebra we cannot decide in which category it sits, until we decide how to view the operad of commutative algebras.

So, there is no *the* operad $(\overline{\mathfrak{G}}, \overline{\mathfrak{R}})$, but many such. Let \mathfrak{H} be a graph-algebra, let $\{\alpha_i : \mathfrak{G}_i \rightarrow \mathfrak{H}\}_{i \in \mathcal{I}}$ be morphisms of graph-algebras, and suppose we have a representation $\mathfrak{R} : \mathfrak{H} \rightarrow \mathcal{M}$, s.t. \mathfrak{R}_i are given by compositions

$$\mathfrak{G}_i \xrightarrow{\alpha_i} \mathfrak{H} \rightarrow \mathcal{M}. \quad (60)$$

When can we regard $(\mathfrak{H}, \mathfrak{R})$ as an operad of operations between \mathfrak{R}_i ’s?

The first condition is that each \mathfrak{G}_i ’s is a **full sub-category** in \mathfrak{H} , i.e. both components of the double functor $\alpha_i : \mathfrak{G}_i \rightarrow \mathfrak{H}$ are full and faithful, and for any $\phi \in \mathfrak{M}_{\mathfrak{H}}$ s.t. $\sigma(\phi), \tau(\phi) \in \alpha_i(\mathfrak{G}_i)$ also $\phi \in \alpha_i(\mathfrak{G}_i)$. We will call such α_i ’s **embeddings**.

The second condition is that objects of \mathfrak{H} should represent operations between objects of $\alpha_i(\mathfrak{G}_i)$'s. Let $\mathcal{O} \subset \text{Obj}(\mathfrak{D}_{\mathfrak{H}})$ be a set of graphs in \mathfrak{H} . We will say that a graph $\mathcal{H} \in \text{Obj}(\mathfrak{D}_{\mathfrak{H}}) - \mathcal{O}$ is a **boundary graph** of \mathcal{O} , if for every non-invertible morphism $\mathcal{H} > \mathcal{G}$ in $\mathfrak{D}_{\mathfrak{H}}$, we have $\mathcal{G} \in \mathcal{O}$.

We will say that a full sub-category of $\mathfrak{H}' < \mathfrak{H}$ is **closed** in \mathfrak{H} , if its boundary is empty. If $\mathfrak{H}' < \mathfrak{H}$ is not closed, its **closure** $\overline{\mathfrak{H}'} < \mathfrak{H}$ is the smallest closed full sub-category of \mathfrak{H} , containing \mathfrak{H}' .

Now we can say, that $(\mathfrak{H}, \mathfrak{R})$ as above is **an operad of operations between operads** $\{(\mathfrak{G}_i, \mathfrak{R}_i)\}_{i \in \mathcal{I}}$, if every α_i is an embedding, and closure of the image of

$$\coprod_{i \in I} \mathfrak{G}_i \rightarrow \mathfrak{H}, \quad (61)$$

is all of \mathfrak{H} .

4 Examples

Now we come to the important concept of a weak higher category. Different from many approaches, we do not want to give *the* definition of weak higher categories, since in our opinion it involves too many choices, that might be valid only in some of the situations in practice. Instead, we view the notion of a weak category as relative to the class of graph-algebras chosen.

Some choices might lead to equivalent definitions, others might not, and all of it depends, of course, on the definition of “equivalence”. However, in this paper proving existence of equivalence is not as central as finding ways to translate from one approach to another. This ability to switch between different combinatorial models for higher categories was the main goal for this work.

In our language a category is a \mathfrak{G} -operad, for some special graph-algebra \mathfrak{G} , and therefore, to define weak categories we have to understand what are weak \mathfrak{G} -operads, for a given graph-algebra \mathfrak{G} .

There is a standard answer: a weak \mathfrak{G} -operad is a representation of a graph-algebra \mathfrak{H} , s.t. there is a *weak equivalence* of graph algebras $\mathfrak{H} \rightarrow \mathfrak{G}$. Of course, we have to define first which morphisms of graph-algebras are weak equivalences.

There are many different ways to define weak equivalences between graph-algebras. For example one can compare their nerves (which are category objects in simplicial sets) or nerves of some subcategories in $\mathfrak{M}_{\mathfrak{G}}, \mathfrak{M}_{\mathfrak{H}}$. Here the resulting homotopy theories of nested operads are rather crude.

A finer definition, used in the globular approach to weak higher categories ([Ba98]), has the following type: a morphism of graph algebras $\mathfrak{F} : \mathfrak{H} \rightarrow \mathfrak{G}$ is a weak equivalence, if the functor between categories of objects is an equivalence of categories, and \mathfrak{F} has the right lifting property with respect to some class of morphisms between double categories $\{F_i : \mathcal{U}_i \rightarrow \mathcal{V}_i\}_{i \in I}$, s.t. for each i the functor between categories of objects, given by F_i , is an isomorphism.

To make this definition type into an actual definition, one has to choose F_i 's, but their choice depends on the particular kind of graph-algebras $\mathfrak{G}, \mathfrak{H}$ under consideration. For example, if \mathfrak{G} is the graph-algebra, whose representations are strict globular categories, F_i 's are defined to ensure that if \mathfrak{H} has two pastings for the same pasting diagram, they are connected by a higher morphism.

This type of definitions of weak equivalences is, in a sense, a generalization of the notion of weak equivalences between simplicial categories. Indeed, the latter consists of requiring that a simplicial functor $\mathfrak{F} : C_1 \rightarrow C_2$ induces an equivalence of categories of components $\pi_0(C_1) \rightarrow \pi_0(C_2)$, and for each pair of objects in C_1 , the space of morphisms is weakly equivalent to the space of morphisms between the images in C_2 .

We will not try to give a precise definition of a weak equivalence between general graph-algebras, nor will we investigate possible models for the resulting homotopy theory. Instead we will consider some special examples, and in the second paper ([Bo09]) we will compare them, and describe translations from one to another.

All of the examples that we will consider in this section describe oriented categories. Here by orientation we mean that morphisms have sources and targets, i.e. orientations.

Notice, that nowhere so far have we assumed that nested graphs under consideration are oriented. Nodes in nested graphs represent operations, but a priori there is no division of flags into incoming and outgoing.

So, orientation is a particular kind of labeling (just as in the theory of plain graphs), and we investigate oriented graphs in the first part of this section.

After that we consider two particular examples: weak globular and weak cubical categories. The former was developed by M.Batanin in [Ba98], and the latter is based on the notion of strict cubical categories ([ABS02], and references therein).

It was proved in [ABS02], that strict globular categories are equivalent to strict cubical ones, and this is the basis for equivalence of the weak ver-

sions. However, we will need some additional machinery (construction of free \mathfrak{G} -operads, for a given graph-algebra \mathfrak{G}), and we postpone the actual comparison until [Bo09].

In the last part of this section we develop the general *slice construction*. Slice construction for colored May operads is the basis for the Baez-Dolan approach to weak higher categories ([BD98]), and we generalize it to arbitrary graph-algebras. That is, we show how, starting with a graph-algebra \mathfrak{G} , to build a new graph-algebra \mathfrak{G}^+ , obtained from \mathfrak{G} by turning compositions of operations into operations, laws for compositions of operations into compositions of operations and so on.

The generalization of the slice construction that we provide is more involved than the original one, not only because one can apply it to graph-algebras and operads of any combinatorics (e.g. cubical), but also because it has additional components, reflecting the fact that we deal with nested operads, and not merely colored May ones.

As a result we can obtain Batanin's theory, as well as the Baez-Dolan one, starting with the trivial operad and iterating slice construction. This leads to comparison between theories, that will be done in [Bo09].

4.1 Oriented graphs

So far all of our graphs were non-oriented. We have used categories to represent nested graphs, and morphisms in such categories obviously have direction, however, it describes hierarchy of dependencies between operations, and not orientation on edges and stars of the graphs.

Now we restrict our attention to *oriented* nested graphs, that we introduce in the following definition.

Definition 29 *An oriented nested graph is a nested graph \mathcal{G} , together with an orientation on some flags (i.e. choice of one of the two adjacent nodes as source, and the other as target, unrelated to direction of the flag as a morphism), s.t.*

1. *All irreducible flags are oriented.*¹⁶
2. *If two flags $f, g \in \mathcal{G}$ are composable, and are oriented in a compatible way, then their composition is also oriented in the same direction.*
3. *If a flag $f \circ g \in \mathcal{G}$ is oriented, then both f, g are oriented in the same direction as $f \circ g$.*

¹⁶Recall that a flag is irreducible if it cannot be obtained as a composition of 2 flags.

Notice that we do not assume absence of oriented cycles. If we impose such conditions, we would not be able to define contractions between oriented graphs, since in our definition contractions do not contract edges, but only loops.

It is easy to see that an orientation on a graph is completely determined by orientations of the irreducible flags.

We will say that a functor $\phi : \mathcal{G} \rightarrow \mathcal{H}$ between oriented nested graphs **preserves orientation**, if for any flag $f \in \mathcal{H}$, s.t. all of its pre-images in \mathcal{G} are oriented in a compatible way, f is oriented in the same direction.

Note, that this definition implies that dependencies and mergers preserve orientation if and only if they map oriented flags to oriented flags, and the orientations agree. The condition on contractions to preserve orientation is more complicated, since contractions can identify some flags.

We define dependencies, mergers, and contractions between oriented nested graphs as respectively dependencies, mergers, contractions between the underlying nested graphs, that preserve orientation. In this way we get the category \mathfrak{D} of oriented nested graphs, and the category \mathfrak{M} of morphisms, where objects are mergers/contractions between oriented nested graphs, and morphisms are triples of dependencies.

The following proposition shows that the double category structure on \mathfrak{N} induces a double category structure on $\mathfrak{N} := (\mathfrak{D}, \mathfrak{M})$.

Proposition 26 *Compositions of morphisms of oriented graphs are morphisms of oriented graphs.*

Proof: It is clear that compositions of oriented mergers are oriented mergers, and compositions of oriented contractions are oriented contractions. Let $\mathcal{G}_1 \xrightarrow{\kappa} \mathcal{G}_2 \xrightarrow{\mu} \mathcal{G}_3$ be an oriented contraction and an oriented merger respectively. We claim that $(\mu', \kappa') := \gamma(\mu, \kappa)$, as defined by (13), is a morphism of oriented graphs, i.e. both μ' and κ' are oriented.

To prove it, we have to first define an orientation on the codomain \mathcal{G}'_2 of μ' . By definition of μ' in (13) \mathcal{G}'_2 is given by identifying some nodes in \mathcal{G}_1 , and flags of \mathcal{G}'_2 are freely created by flags of \mathcal{G}_1 , subject to relations of composition in \mathcal{G}_1 . It is clear that there is a unique orientation on \mathcal{G}'_2 , that extends the orientation on the image of μ' and is smallest such. Clearly μ' is an oriented merger.

It is also easy to see that κ' has to be an oriented contraction, since the composite functor $\mu \circ \kappa$ preserves orientation. ■

It is straightforward to prove that the partial \mathfrak{S} -category structure on \mathfrak{N}^{op} , defined in section 3.3, extends to $\mathfrak{N}^{op}_{\rightarrow}$, such that the forgetful functor $\mathfrak{N}^{op}_{\rightarrow} \rightarrow \mathfrak{N}^{op}$ is a morphism of \mathfrak{S} -categories. Clearly $\mathfrak{N}_{\rightarrow}$ is a graph-algebra.

Direction is one example of labeling, that is used very often in the theory of operads and categories. There is another labeling, that is not general, but specific to our definitions of nested graphs and algebraic operations with them.

As we have defined them, nested graphs are categories, where objects represent nodes and morphisms represent flags. A contraction from one nested graph to another contracts some flags, and we have imposed a condition on these contractions: they have to be direct functors, which means that if a flag is contracted, and another flag is decorated by the same node, then the other flag has to be contracted as well, or at least identified with a flag, whose decoration is not contracted.

In other words, if two flags are decorated by the same node, they participate together in all contractions. However, sometimes we would like one object to participate in several compositions, each one of them being executed separately, as in the following example. Consider the operation of horizontal composition:

$$A \begin{array}{c} \xrightarrow{f_1} \\ \Downarrow \phi \\ \xrightarrow{g_1} \end{array} B \begin{array}{c} \xrightarrow{f_2} \\ \Downarrow \psi \\ \xrightarrow{g_2} \end{array} C \mapsto A \begin{array}{c} \xrightarrow{f_2 f_1} \\ \Downarrow \psi \phi \\ \xrightarrow{g_2 g_1} \end{array} C \quad (62)$$

The left part of (62) is naturally represented by a direct category as follows:

$$\begin{array}{ccccc} & & \phi & & \psi \\ & \nearrow & & \nwarrow & \nearrow \\ f_1 & & & & f_2 \\ & \nwarrow & & \nearrow & \nwarrow \\ & A & & B & & C \\ & \nearrow & & \nwarrow & \nearrow & \nwarrow \\ & & g_1 & & g_2 \end{array} \quad (63)$$

and the right part is

$$\begin{array}{ccc} & \psi \phi & \\ \nearrow & & \nwarrow \\ f_2 f_1 & & g_2 g_1 \\ \nearrow & & \nwarrow \\ A & & C \end{array} \quad (64)$$

However, there is no functor, going from (63) to (64), that represents horizontal composition, because B has to be contracted both into $f_2 f_1$ and $g_1 g_1$, which is impossible.

A natural way around this difficulty would be to take two copies of B in (63), one for the edge f_1, g_1 , and the other for f_2, g_2 . However, we would like to have both B 's standing for the same set of objects, i.e. we should remember that they are just copies. We do this by having an equivalence relation on the nodes in a nested graph, and then requiring that representations of equivalent elements are equal.

Definition 30 *An equivalence relation \sim on a nested graph \mathcal{G} is a pair of equivalence relations: one on nodes, and one on flags, s.t.*

1. *The relations are compatible with the categorical structure, i.e. the domain, codomain maps $Mor(\mathcal{G}) \rightrightarrows Obj(\mathcal{G})$, and the composition map $Mor(\mathcal{G}) \times_{Obj(\mathcal{G})} Mor(\mathcal{G}) \rightarrow Mor(\mathcal{G})$ are \sim -invariant,*
2. *the category \mathcal{G}/\sim is a direct category.*

It is clear, that we can extend the notion of equivalence relations from arbitrary nested graphs to oriented ones, by requiring that equivalences are compatible with orientations, in the obvious sense.

A functor ϕ between nested graphs with equivalence relations is **compatible with equivalence relations** if $A \sim B$ implies $\phi(A) \sim \phi(B)$ or at least one of A, B is contracted by ϕ , and $f \sim g$ implies $\phi(f) \sim \phi(g)$ or at least one of f, g is contracted by ϕ .

We can define mergers, contractions and dependencies between oriented graphs with equivalences, and it is clear that we have a new graph-algebra, that we will denote again by \mathfrak{N} .

Let $\delta_1, \delta_2 : \mathcal{G} \rightarrow \mathcal{H}$ be two dependencies between nested graphs with equivalences. We will say that $\delta_1 \sim \delta_2$, if for any node $A \in \mathcal{H}$, $\delta_1(A) \sim \delta_2(A)$, and similarly for flags. It is clear that in this way we get an equivalence relation on the set of dependencies between nested graphs with equivalences, and in particular on $Mor(\mathfrak{D})$ and $Mor(\mathfrak{M})$.

Now, having \mathfrak{S} -categories with equivalences on morphisms, both in the category of objects, and category of morphisms, we define a morphism of such \mathfrak{S} -categories as a morphism, that preserves equivalence relations.

In particular we can apply it to representations of graph-algebras with equivalences. Let \mathcal{M} be an \mathfrak{S} -algebra. The corresponding \mathfrak{S} -category

$(\mathcal{M}, Mor(\mathcal{M}))$ has the trivial equivalence relations, and hence an operad $\mathfrak{N} \rightarrow (\mathcal{M}, Mor(\mathcal{M}))$ has the property that images of equivalent dependencies are equal.

4.2 Examples of oriented categories

In this section we consider two examples of applications of our constructions. First we show that globular operads, and weak globular categories, developed in [Ba98], can be obtained as a particular case of our constructions.

Then we develop the theory of cubical operads, based on the notion of strict cubical categories ([ABS02], and references therein).

4.2.1 Globes

Here we present the theory of weak globular categories, formulated in the language of graph-algebras and their representations.

This theory is based on the notion of *strict globular categories* which are just the usual strict higher categories. The combinatorics of these categories is given by globes of dimensions ≥ 0 , where an n -dimensional globe is obtained from an n -dimensional ball, whose boundary is divided into two $n - 1$ -dimensional balls, one incoming and one outgoing, that intersect on their boundaries, and so on.

One can glue an ordered pair of n -dimensional balls in n -different ways, and these operations satisfy the usual associativity and interchange axioms of strict higher categories. In addition one can consider an $n - 1$ -ball as a degenerate n -ball, i.e. there is a degeneracy operation, that is right inverse to taking the boundary.

One can attempt to describe this structure in terms of nested graphs as follows: starting with an n -ball, and taking all m -balls, $m \leq n$, in the boundary as objects, and inclusions as morphisms, we get the direct category $\underline{\Phi}_n$, having $2n + 1$ objects and obvious morphisms.

It is immediately clear that $\underline{\Phi}_n$'s are not compatible with our language of describing operations as direct functors between direct categories. Consider the example of gluing two copies of $\underline{\Phi}_2$ along $\underline{\Phi}_0$. This is the situation of (62), and as it was explained there, to describe such gluing in terms of nested graphs, one has to take copies of nodes and introduce equivalence relations.

This is what we are going to do now. As a motivation, consider $\underline{\Phi}_2$. It has two nodes representing objects, two nodes standing for 1-morphisms, and one node representing a 2-morphism. To be able to describe horizontal

products by direct functors, we have to take an additional copy for each one of the object-nodes. Graphically this looks as follows:

$$\cdot \begin{array}{c} \curvearrowright \\ \Downarrow \\ \curvearrowleft \end{array} \cdot \quad \longrightarrow \quad : \begin{array}{c} \curvearrowright \\ \Downarrow \\ \curvearrowleft \end{array} : \quad (65)$$

On the right hand side we get a direct category Φ_2 having 2^2 nodes representing objects, 2^1 -nodes standing for 1-morphisms, and 2^0 nodes representing 2-morphisms. Performing this procedure for a general $\underline{\Phi}_n$ we get a direct category Φ_n having, for each $0 \leq k \leq n$, 2^{n-k} nodes representing k -morphisms. In total Φ_n has $2^{n+1} - 1$ nodes.

Here are the precise definitions. Let Φ_n be a direct category defined as follows. The set $Obj(\Phi_n)$ has $2^{n+1} - 1$ elements, and it is a disjoint union of subsets $\{V_i\}_{0 \leq i \leq n}$, s.t. V_i has 2^i elements. We assume that each V_i is totally ordered, and we number elements of V_i with $1, \dots, 2^i$, starting with the smallest. We will use the even-numbered nodes as outputs, and the odd-numbered ones as inputs.

To define morphisms we consider the (weakly) order preserving maps $\partial_i : V_i \rightarrow V_{i-1}$, s.t. every fiber of every ∂_i has precisely 2 elements. It is easy to see that for each $i > 0$ there is exactly one such ∂_i .

If for $v_i \in V_i$ we have $\partial_i(v_i) = v_{i-1} \in V_{i-1}$, then we have a morphism $v_i \rightarrow v_{i-1}$. By definition of ∂_i , there are two elements v_i, v'_i that are mapped to v_{i-1} . The morphism from the smaller one (according to the linear order on V_i) we will denote by ∂_i^0 , and morphism from the larger one by ∂_i^1 .

We define Φ_n to have $\coprod_{0 \leq i \leq n} V_i$ as objects and having morphisms freely generated by $\partial_i^0, \partial_i^1$'s. There is an orientation on Φ_n , where we define ∂_i^0 as being the input flag for v_{i-1} , and ∂_i^1 as the output flag.

We also define an equivalence relation, by declaring two nodes to be equivalent, if they are in the same V_i , and their numbers have the same parity. We say that two irreducible flags are equivalent if they connect pairwise equivalent elements, and we complete the definition by requiring that if $f \sim g$, then $h \circ f \sim h \circ g$ for any h .

It is easy to see that if we identify equivalent objects and morphisms in Φ_n , we will get $\underline{\Phi}_n$. Hence, Φ_n is an oriented nested graph with equivalences.

We allow all dependencies $\Phi_m \rightarrow \Phi_n$, and in this way we get the category of corollas. The rest of the category of objects \mathfrak{D}_{Φ} and the category of morphisms \mathfrak{M}_{Φ} are obtained in two steps.

Consider first two copies Φ_n, Φ'_n . If we identify equivalent nodes and flags, we obtain $\underline{\Phi}_n, \underline{\Phi}'_n$, and it is well known what it means to glue the

latter two along a k -dimensional boundary. We consider only those gluings that identify an output face of Φ'_n with an input face of Φ_n .

If we don't identify equivalent nodes and flags in Φ_n, Φ'_n , we still can perform the same gluing along a k -dimensional boundary, but now there are 2^{n-k-1} copies of this boundary in each one of Φ_n, Φ'_n . We glue pairwise, preserving the order relations on the copies.

Iterating this procedure, we get graphs, representing all possible pasting diagrams, where in each diagram the dimensions of participating globes are constant. To get diagram with some of the globes being degenerate, we have to perform the second step.

Now we define the category of morphisms \mathfrak{M}_Φ . First we take all non-degenerate pastings, i.e. having an object $\mathcal{G} \in \mathfrak{O}_\Phi$, defined in the first step, we introduce a map $\mathcal{G} \rightarrow \Phi_n$, where n is the common dimension of the globes in \mathcal{G} .

In addition to the pastings, we define degeneracy morphisms $\Phi_n \rightarrow \Phi_{n+1}$, that are right inverse to boundaries $\Phi_{n+1} \rightarrow \Phi_n$. Now we complete the definition of $\mathfrak{M}_\Phi, \mathfrak{O}_\Phi$, by taking all compositions of morphisms, and all \mathfrak{S} -products, subject to the usual associativity/interchange/identities laws of pastings in strict higher categories.

Here we have to add new objects to \mathfrak{O}_Φ , since, for example, composition of the degeneracy $\Phi_1 \rightarrow \Phi_2$, and the gluing $\Phi_2 * \Phi_2 \rightarrow \Phi_2$ is the gluing $\Phi_2 * \Phi_1 \rightarrow \Phi_2$.

Since we defined $\mathfrak{O}_\Phi, \mathfrak{M}_\Phi$ by taking closure with respect to compositions and \mathfrak{S} -structure, starting with corollas and constants, it is clear that the resulting $\Phi := (\mathfrak{O}_\Phi, \mathfrak{M}_\Phi)$ is a graph-algebra.

Now let $\mathfrak{F} : \tilde{\Phi} \rightarrow \Phi$ be a morphism of graph algebras, s.t. $\mathfrak{O}_{\tilde{\Phi}} \rightarrow \mathfrak{O}_\Phi$ is an isomorphism of categories, and $\mathfrak{M}_{\tilde{\Phi}} \rightarrow \mathfrak{M}_\Phi$ is a full and faithful functor.

We define $\mathcal{P}, \tilde{\mathcal{P}}$ to be the sets of isomorphism classes of those objects in $\mathfrak{M}_\Phi, \mathfrak{M}_{\tilde{\Phi}}$ respectively, whose codomains are corollas. We have a natural grading on both \mathcal{P} and $\tilde{\mathcal{P}}$ by dimension of the codomain. We will denote by $\mathcal{P}_n, \tilde{\mathcal{P}}_n$ the sets of objects of degree n in $\mathcal{P}, \tilde{\mathcal{P}}$ respectively.

In the following proposition we show that $\tilde{\Phi}$ as above determines an ω -globular operad, as defined in [Ba98], and representations of $\tilde{\Phi}$ correspond to algebras over this operad. In its formulation and proof, we use freely the terminology of [Ba98].

Proposition 27 *Let $\tilde{\Phi}$ be as above. Then*

1. *Dependencies and constants make $\mathcal{P} = \{\mathcal{P}_n\}_{n \geq 0}$ into a discrete globular category, and the \mathfrak{S} -category structure on Φ makes \mathcal{P} into a*

monoidal globular category, that is isomorphic to the category of many-stage trees Tr .

2. By dividing $\tilde{\mathcal{P}} = \{\tilde{\mathcal{P}}_n\}_{n \geq 0}$ into fibers of \mathfrak{F} , we realize $\tilde{\mathcal{P}}$ as a collection in the monoidal globular category $Span$ of spans, and the \mathfrak{S} -structure on $\tilde{\Phi}$ makes $\tilde{\mathcal{P}}$ into a monoidal globular operad.
3. Let $\mathfrak{R} : \tilde{\Phi} \rightarrow Set$ be a representation of $\tilde{\Phi}$. Then \mathfrak{R} restricted to corollas in $\mathfrak{D}_{\tilde{\Phi}}$ defines a globular object in $Span$, and the rest of \mathfrak{R} defines the structure of a $\tilde{\mathcal{P}}$ -algebra on this globular object.
4. Conversely, every monoidal globular operad \tilde{P} in $Span$, and every \tilde{P} -algebra in $Span$ can be realized in this way as representation of a suitable $\tilde{\Phi}$ in Set .

Proof:

1. Dependencies in \mathfrak{M} are completely determined by their projection to \mathfrak{D}_{Φ} , given by the target map τ . Since targets of objects in \mathcal{P} are corollas, and corollas in \mathfrak{D}_{Φ} are globes of all dimensions, it is clear that dependencies in \mathcal{P} satisfy the standard axioms of boundaries in globular combinatorics.

We have added constants $\Phi_n \rightarrow \Phi_{n+1}$, postulating that they are the right inverses for the boundaries, and moreover, they are supposed to have the properties of identities in globular combinatorics. So it is clear that \mathcal{P} is a globular category.

To identify \mathcal{P} with Tr we note, that the source map $\sigma : \mathfrak{M}_{\Phi} \rightarrow \mathfrak{D}_{\Phi}$ induces a surjective map $\mathcal{P} \rightarrow Obj(\mathfrak{D}_{\Phi})$. Now objects of \mathfrak{D}_{Φ} are precisely direct categories, that are diagrams of multi-stage trees, so each object in \mathcal{P} corresponds to a multi-stage tree.

Moreover, different elements of a fiber of $\mathcal{P} \rightarrow Obj(\mathfrak{D}_{\Phi})$ correspond to different number of empty “layers” added to that tree. So we see that Tr and \mathcal{P} are isomorphic as globular categories.

Finally, the globular monoidal structure on Tr is given by gluing trees over sub-trees, or in our language by gluing graphs over full subgraphs. It is easy to see that these operations in Tr and \mathcal{P} agree.

2. Dividing $\tilde{\mathcal{P}}$ into fibers over \mathcal{P} we obtain an assignment of a set (fiber) to each element of \mathcal{P} , and since the functor $\mathfrak{M}_{\tilde{\Phi}} \rightarrow \mathfrak{M}_{\Phi}$ is assumed

to be full and faithful, this assignment is in fact a functor from Tr to globular sets, i.e. $\tilde{\mathcal{P}}$ can be seen as a collection in $Span$.

Since $\tilde{\Phi}$ is a graph-algebra, we know how, starting with a diagram of elements of $\tilde{\mathcal{P}}$, to compose these elements into a object $\phi \in \mathfrak{M}_{\tilde{\Phi}}$, whose codomain is not necessarily a corolla, and then we can compose ϕ with an element of $\tilde{\mathcal{P}}$, whose domain is the codomain of ϕ .

It is straightforward to check that these compositions define the structure of a globular operad on $\tilde{\mathcal{P}}$.

3. It is clear that \mathfrak{R} , restricted to corollas, defines a globular object in $Span$ (simply because the category of corollas, dependencies and constants is the final globular category).

An action of $\tilde{\mathcal{P}}$ on a globular object is a representation of $\tilde{\mathcal{P}}$ in the corresponding endomorphism operad. It is easy to see that the endomorphism operad for the globular object in $Span$, given by restricting \mathfrak{R} to corollas, is $\{Hom(\mathfrak{R}(\mathcal{G}), \mathcal{C})\}_\phi$, where $\phi : \mathcal{G} \rightarrow \mathcal{C}$ is an element of \mathcal{P} , and the claim is obvious.

4. Let \tilde{P} be any globular operad. We define the category $\tilde{\mathcal{P}}$ in the obvious way (using globular source/target maps to define morphisms, and degeneracies to define additional objects), and we have a functor $\tilde{\mathcal{P}} \rightarrow \mathcal{P}$, sending every element to the tree, it is parameterized with.

Now we generate a partial \mathfrak{S} -algebra by taking all allowed products (orientation, etc.), and dividing by relations of compositions in the operad \tilde{P} . It is straightforward to check that we get a graph-algebra, and that the two operations are inverses for each other.

■

Having defined globular operads, we introduce weak equivalences between them. Let Λ be the discrete double category, having two objects A, B , and only two non-trivial morphisms $A \rightrightarrows B$.

Let Δ be the following double category. The category of objects has 3 objects A, B, C , and three non-trivial morphisms: $B \rightarrow C$, and $C \rightrightarrows B$, s.t. the former is the right inverse to both of the latter. The category of morphisms has 3 objects: $A \rightrightarrows B$, and $A \rightarrow C$, and 2 morphisms from the latter to the two former ones.

Following [Ba98], we say that $\tilde{\Phi}$ is **contractible**, if $\tilde{\Phi} \rightarrow \Phi$ has the right lifting property with respect to the inclusion $\Lambda \rightarrow \Delta$. Then a **weak globular category** is a representation of a contractible $\tilde{\Phi}$ in Set .

4.2.2 Cubes

Here we present the theory of cubical operads. It is based on the graph-algebra of strict cubical categories (for the latter see [ABS02], and references therein).

Different from the globular case, the straightforward combinatorial model of an n -cube is compatible with the theory of nested graphs, i.e. we do not have to use equivalences on nodes. Let Ξ_n be the nested graph, obtained by taking one node for each k -face of the n -cube, for each $k \geq 0$, and taking a flag for each inclusion of a k -face into an m -face, $m > k$.

We assume that all of our cubes are oriented in the standard way, i.e. we choose one vertex in the n -cube as the input. Then the opposite end of the n -diagonal is declared as the output. This defines an orientation on Ξ_n as follows.

An $n - 1$ -face is an input face if it contains the input vertex, and it is an output face if it contains the output vertex. It is clear that in this way we cover all $n - 1$ -faces. Now every $n - 1$ is itself oriented, since it either already has the input vertex, or it has the output one, and then we choose the other end of the $n - 1$ -diagonal as the input.

Continuing like this we assign orientation to every irreducible flag in Ξ_n , and they generate an orientation on all of Ξ_n . It is straightforward to check that orientations on intersections of $n - 1$ -faces are compatible.

We have defined $\{\Xi_n\}_{n \geq 0}$, and these are the corollas in our category of objects \mathfrak{O}_{Ξ} . As in the globular case, we define the rest of the objects in \mathfrak{O}_{Ξ} in two steps.

First, for each $n \geq 0$, we take all possible cubical pasting diagrams, consisting of n -cubes, i.e. copies of Ξ_n . Each such diagram can be thought of as cutting an n -cube into smaller n -cubes, s.t. the cuts are parallel to the sides of the big cube.

Then we add one object ϕ_n to \mathfrak{M}_{Ξ} , for each pasting diagram as above. The source of ϕ_n is the pasting diagram itself, the target is Ξ_n .

In the second step we add the degeneracy morphisms. Different from the globular case, for each $n \geq 0$ there is not one, but $(n + 1)^2$ degeneracies $\Xi_n \rightarrow \Xi_{n+1}$. Indeed, there are $n + 1$ usual ones, i.e. choosing two opposite n -faces in an $n + 1$ -cube, and there are also *connections* ([ABS02]): choosing two adjacent n -faces at the input, or at the output, altogether $n(n + 1)$ choices.

Finally, we define $\Xi := (\mathfrak{O}_{\Xi}, \mathfrak{M}_{\Xi})$ by taking all compositions of objects in \mathfrak{M}_{Ξ} and \mathfrak{S} -products, corresponding to cubical pasting diagrams. This

forces us to add some objects to $\mathfrak{D}\Xi$, consisting of pasting diagrams, where some n -cubes can be degenerate (in either one of n^2 ways).

The result is clearly a graph-algebra, and representations of Ξ in *Set* are precisely the strict cubical categories. Now we can define cubical operads, in analogy to globular ones.

Definition 31 *A cubical operad is a representation $\tilde{\Xi} \rightarrow \text{Set}$, where $\tilde{\Xi}$ is a graph algebra, having a morphism $\mathfrak{F} : \tilde{\Xi} \rightarrow \Xi$, s.t. the functor between categories of objects is an isomorphism, and the one between categories of morphisms is a full and faithful functor.*

It is known that strict cubical categories are equivalent to strict globular ones ([ABS02]). In our language it means that Φ and Ξ are Morita equivalent. This equivalence can be manifested by a kind of Morita context: we can define a graph algebra $\Phi + \Xi$, that contains copies of Φ and Ξ , and is generated by them and some additional morphisms.

To be precise, we should not take Ξ_n 's themselves, but their equivalent versions, where we add several copies of each k -face for each $k \geq 0$, so that the result will be compatible with Φ_n 's. It is straightforward, but tedious, to describe the needed additions, and we will not do it here. We will denote the resulting combinatorial models for n -cubes again by Ξ_n 's.

The additional morphisms in $\Phi + \Xi$ consist of the obvious contractions

$$\Xi_n \rightarrow \Phi_n, \quad (66)$$

which “smooth out” corners, and the inclusions (which we consider as objects in $\mathfrak{M}_{\Phi+\Xi}$, i.e. as constants)

$$\Phi_n \rightarrow \Xi_n \quad (67)$$

that put the two $n - 1$ -globes as opposite faces.

The morphisms from (66), (67) provide the tools to prove Morita equivalence of Φ and Ξ (and they are used in [ABS02]), but they also provide a way to translate cubical operads into globular ones, and vice versa. We will describe this explicitly in [Bo09].

4.3 The slice construction and opetopes

In [BD98] the *slice construction* was defined for colored May operads. Starting with such operad P , its slice construction P^+ is obtained by taking operations of composition of elements in P as elements of P^+ , and defining compositions of elements in P^+ by using relations on compositions in P .

Here we outline a generalization of slice construction to graph-algebras and nested operads. Presenting all the details is rather long, and we postpone it until [Bo09].

Let $\mathfrak{G} = (\mathfrak{D}_{\mathfrak{G}}, \mathfrak{M}_{\mathfrak{G}})$ be some graph-algebra, whose objects are oriented nested graphs (and possibly with other labelings).¹⁷ We are going to define a new graph algebra \mathfrak{G}^+ , that we will call **the slice construction** of \mathfrak{G} . Operads, given as representations of \mathfrak{G}^+ , are the slice constructions of operads, defined by \mathfrak{G} . There are two steps.

1. First we add an object to $\mathfrak{D}_{\mathfrak{G}}$ for every object of $\mathfrak{M}_{\mathfrak{G}}$, i.e. we turn \mathfrak{G} -compositions into \mathfrak{G}^+ -operations. This is a straightforward generalization of the similar part in the original slice construction ([BD98]).

Let $\phi : \mathcal{G}_1 \rightarrow \mathcal{G}_2$ be an object of $\mathfrak{M}_{\mathfrak{G}}$ (i.e. ϕ is an oriented merger, followed by an oriented contraction). We define a new oriented graph \mathcal{G}_{ϕ} as follows.

The construction is similar to the cone of a morphism in homological algebra. We take one copy of \mathcal{G}_1 , and two copies $\mathcal{G}_2, \mathcal{G}'_2$ of \mathcal{G}_2 , and we define \mathcal{G}_{ϕ} by adding some flags to $\mathcal{G}_1 \coprod \mathcal{G}_2 \coprod \mathcal{G}'_2$. The idea is that every node $A' \in \mathcal{G}'_2$ should represent the morphism from $\phi^{-1}(A)$ to A , given by ϕ , where A is the copy of A' in \mathcal{G}_2 .

For any node $A \in \mathcal{G}_2$, and its copy $A' \in \mathcal{G}'_2$, we add a flag $f_A : A \rightarrow A'$, and for every node $B \in \phi^{-1}(A) \subseteq \mathcal{G}_1$, we add a flag $f_B : B \rightarrow A'$. Then we take all possible compositions of $\{f_A, f_B\}_{A \in \mathcal{G}_2, B \in \mathcal{G}_1}$ with flags in $\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}'_2$, subject to the following relations.

Let $g : A_1 \rightarrow A_2$ be a flag in \mathcal{G}_2 , and let g' be its copy in \mathcal{G}'_2 . Then we set

$$f_{A_2} \circ g = g' \circ f_{A_1}. \quad (68)$$

Let $h : B_1 \rightarrow B_2$ be a flag in \mathcal{G}_1 , s.t. $g = \phi(h)$, for some morphism $g \in \mathcal{G}_2$ (g can be an identity). Let g' be the copy of g in \mathcal{G}'_2 . We set

$$f_{B_2} \circ h = g' \circ f_{B_1}. \quad (69)$$

It is clear that \mathcal{G}_{ϕ} is a nested graph, and moreover, if \mathcal{G}_2 is a corolla, then \mathcal{G}_{ϕ} is a corolla as well. We define an orientation on \mathcal{G}_{ϕ} by extending the orientations on $\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}'_2$, and declaring that irreducible flags from \mathcal{G}_1 to \mathcal{G}'_2 are incoming, and irreducible flags from \mathcal{G}_2 to \mathcal{G}'_2 are outgoing.

Consider some examples. Let pt be the nested graph, consisting of one node, and let $\phi : pt \rightarrow pt$ be the unique automorphism. Then \mathcal{G}_{ϕ} is just the

¹⁷One can also define slice construction for non-oriented graphs, but we will not need such generality in this paper.

oriented interval. Let $\psi : \mathcal{G}_\phi \rightarrow \mathcal{G}_\phi$ be the unique (oriented) automorphism. Then \mathcal{G}_ψ is the square. Continuing like this we get all cubical corollas.

We can get globular corollas, if we collapse some of the faces of the cubes, or, better, substitute these faces with equivalence relations as in section 4.2.1. If we take ϕ to be a composition of globes, instead of the identity automorphism, and similarly collapse some faces, we get an opetope.

So we can see, that globes and opetopes are obtained by iterative application of the slice construction to the trivial graph, and then collapsing some faces.

So far we have graphs from $\mathfrak{D}_\mathfrak{G}$, and $\{\mathcal{G}_\phi\}_{\phi \in \mathfrak{M}_\mathfrak{G}}$. These are not all of the objects of $\mathfrak{D}_{\mathfrak{G}^+}$. To describe the rest we need to consider dependencies between graphs that we already have.

It is easy to see that given a morphism $\phi > \psi$ in $\mathfrak{M}_\mathfrak{G}$, we have a dependency $\mathcal{G}_\phi > \mathcal{G}_\psi$. We have, of course, the morphisms in $\mathfrak{D}_\mathfrak{G}$. And in addition we have dependencies $\mathcal{G}_\phi > \mathcal{G}_1$, $\mathcal{G}_\phi > \mathcal{G}_2$, where $\phi : \mathcal{G}_1 \rightarrow \mathcal{G}_2$.

Correspondingly there are three types of gluings we can perform: the ones coming from the partial \mathfrak{S} -algebras structure on $\mathfrak{M}_\mathfrak{G}$, the ones coming from the partial \mathfrak{S} -algebra structure on $\mathfrak{D}_\mathfrak{G}$, and the ones given by the composition γ on $\mathfrak{M}_\mathfrak{G}$.

We define $\mathfrak{D}_{\mathfrak{G}^+}$ to be the partial \mathfrak{S} -algebra, generated by the objects, morphisms, and gluings as above.

2. Now we describe $\mathfrak{M}_{\mathfrak{G}^+}$. For every pair $\mathcal{G}_1 \xrightarrow{\phi} \mathcal{G}_2 \xrightarrow{\psi} \mathcal{G}_3$ of composable objects in $\mathfrak{M}_\mathfrak{G}$, we have a merger/contraction $\gamma_{\psi\phi} : \mathcal{G}_\psi * \mathcal{G}_\phi \rightarrow \mathcal{G}_{\psi\circ\phi}$, where $\mathcal{G}_\psi * \mathcal{G}_\phi$ is the result of gluing \mathcal{G}_ϕ , \mathcal{G}_ψ over \mathcal{G}_2 . We will call such $\gamma_{\psi\phi}$ a **vertical composition**, and we will denote the set of vertical compositions by Γ .

It is easy to see, that using the \mathfrak{S} -algebra structure on $\mathfrak{M}_\mathfrak{G}$ and vertical compositions, we get all possible “operadic” compositions of objects in $\mathfrak{M}_\mathfrak{G}$, i.e. if we apply this to \mathfrak{G} ’s, that parameterize colored May operads, we will get all the opetopes.

However, there are also *horizontal compositions* that we should add to $\mathfrak{M}_{\mathfrak{G}^+}$. While in [BD98] the operads are only colored, our graph algebras describe nested objects. The idea of a nested graph is that, given a corolla \mathcal{C} , parameterizing some operation, the legs of that corolla are operations on their own, i.e. corollas $\{\mathcal{C}_i\}$.

It is important to note that legs of \mathcal{C}_i ’s are also legs of \mathcal{C} , and in particular if we compose a \mathcal{C}_i with some other operation, it should have an effect on \mathcal{C} itself. In terms of objects in $\mathfrak{M}_\mathfrak{G}$ this is described as follows.

Let $\phi : \mathcal{G}_1 \rightarrow \mathcal{G}_2$, $\psi : \mathcal{H}_1 \rightarrow \mathcal{H}_2$ be objects in $\mathfrak{M}_\mathfrak{G}$, and suppose we have

another pair of objects in $\mathfrak{M}_{\mathfrak{G}}$: $\alpha : \mathcal{G}_1 \rightarrow \mathcal{H}_1$, $\beta : \mathcal{G}_2 \rightarrow \mathcal{H}_2$, making up a commutative square, i.e.

$$\gamma(\beta, \phi) = \gamma(\psi, \alpha). \quad (70)$$

We consider this square as an operation of composition in $\mathfrak{M}_{\mathfrak{G}}$, and consequently represent it by a merger/contraction $D_{\alpha, \beta} : \mathcal{G}_{\phi} \rightarrow \mathcal{G}_{\psi}$, that is an object in $\mathfrak{M}_{\mathfrak{G}^+}$. We will call such D a **horizontal composition**. We will denote the set of horizontal compositions by Δ .

Consider an example: if $\phi : pt \rightarrow pt$ is the unique automorphism of the trivial nested graph, let \mathcal{G} be the result of gluing two copies of \mathcal{G}_{ϕ} at a point, and let $\psi : \mathcal{G} \rightarrow \mathcal{G}$ be the identity automorphism. And let $\nu : \mathcal{G}_{\phi} \rightarrow \mathcal{G}_{\phi}$ be the identity automorphism.

We have a commutative square

$$\begin{array}{ccc} \mathcal{G} & \xrightarrow{\psi} & \mathcal{G} \\ \gamma * \gamma \downarrow & & \downarrow \gamma * \gamma \\ \mathcal{G}_{\phi} & \xrightarrow{\nu} & \mathcal{G}_{\phi} \end{array} \quad (71)$$

where $\gamma : \mathcal{G} \rightarrow \mathcal{G}_{\phi}$ is the composition, and $\gamma * \gamma$ is the gluing of two copies of γ . It is clear that $D_{\gamma * \gamma, \gamma * \gamma} : \mathcal{G}_{\psi} \rightarrow \mathcal{G}_{\phi}$ is the usual (cubical) horizontal composition.

Now we define $\mathfrak{M}_{\mathfrak{G}^+}$ to be generated by $\mathfrak{M}_{\mathfrak{G}}$, Γ , and Δ by taking \mathfrak{S} -products and compositions. One has to give, of course, a precise definition of \mathfrak{S} -products allowed, and one has to prove that the result is a graph-algebra (i.e. a partial \mathfrak{S} -category). This is rather long, and it will be done in [BD98].

There is an addition to \mathfrak{G}^+ , that one might want to make. It corresponds to the choice of *universal compositions* in [BD98]. Let $\phi : \mathcal{G}_1 \rightarrow \mathcal{G}_2$ be an object in $\mathfrak{M}_{\mathfrak{G}}$, and let \mathcal{G}_{ϕ} be the corresponding object in $\mathfrak{D}_{\mathfrak{G}^+}$. There is an obvious dependency $\mathcal{G}_{\phi} > \mathcal{G}_1$, and one can add the right inverse to it, as a new object $\mathcal{G}_1 \rightarrow \mathcal{G}_{\phi}$ in $\mathfrak{M}_{\mathfrak{G}^+}$, i.e. add a constant, as described in section 3.4.

Such constants allow us to connect representations of $\phi \in \mathfrak{M}_{\mathfrak{G}}$ and of $\mathcal{G}_{\phi} \in \mathfrak{D}_{\mathfrak{G}^+}$. Indeed, given a representation \mathfrak{R} of \mathfrak{G}^+ in *Set*, every element $p \in \mathfrak{G}(\mathcal{G}_1)$ defines an element in $\mathfrak{R}(\mathcal{G}_{\phi})$, by applying $\mathfrak{R}(\mathcal{G}_1 \rightarrow \mathcal{G}_{\phi})$ to p .

For example, if ϕ is the identity automorphism of \mathcal{G}_1 , we get in this way the usual degeneracy map. If, on the other hand, ϕ is the composition of 1-dimensional globes, then for every pair $(p, q) \in \mathfrak{R}(\mathcal{G}_1)$ of 1-dimensional globes, we have a chosen element $\mathfrak{R}(\mathcal{G}_1 \rightarrow \mathcal{G}_{\phi})(p, q)$ in the opetope $\mathfrak{R}(\mathcal{G}_{\phi})$.

References

- [ABS02] F.A.Al-Agl, R.Brown, R.Steiner. *Multiple categories: the equivalence of a globular and a cubical approach*. Advances in Mathematics 170, pp. 71-118 (2002).
- [Ba98] M.A.Batanin. *Monoidal globular categories as a natural environment for the theory of weak n -categories*. Advances in Mathematics 136, pp. 39-103 (1998).
- [BD98] J.C.Baez, J.Dolan. *Higher dimensional algebra III. n -Categories and the algebra of opetopes*. Advances in Mathematics 135, pp. 135-206, (1998).
- [BeM96] K.Behrend, Yu. Manin. *Stacks of stable maps and Gromov–Witten invariants*. Duke Math. Journ, vol. 85, No. 1, pp. 1-60 (1996).
- [Bo09] D.Borisov. *Comparing definitions of weak higher categories, II*, in preparation.
- [BoM08] D.Borisov, Yu.I.Manin. *Generalized operads and their inner cohomomorphisms*. In *Geometry and dynamics of groups and spaces*. Progress in Mathematics 265, pp. 247-308, Birkäuser, Basel (2008).
- [Bu93] A.Burroni. *Higher-dimensional word problems with applications to equational logic*. Theoretical computer science, 115, pp. 43-62, (1993).
- [Ch03] E.Cheng. *The theory of opetopes via Kelly-Mac Lane graphs*. arXiv:math.CT/0304288v1
- [Ch04] E.Cheng. *Weak n -categories: opetopic and multitopic foundations*. Journal of Pure and Applied Algebra 186, pp. 109-137 (2004).
- [Ch06] E.Cheng. *A relationship between trees and Kelly-Mac Lane graphs*. Math. Proc. Cambridge Philos. Soc., 141 no. 1, pp. 33-56, (2006).
- [GK98] E.Getzler, M.Kapranov. *Modular operads*. Compositio Math., 110:1, pp. 65-126 (1998).
- [Hi05] Ph.J.Higgins. *Thin elements and commutative shells in cubical ω -categories*. Theory and Applications of Categories, Vol. 14, No. 4, pp. 60-74 (2005).

- [HMP98] C.Hermida, M.Makkai, J.Power. *Higher dimensional multigraphs*. In *Thirteenth Annual IEEE Symposium on Logic in Computer Science*. (Indianapolis, IN, 1998), pp. 199-206, IEEE Computer Soc., Los Alamitos, CA, (1998).
- [HMP00] C.Hermida, M.Makkai, J.Power. *On weak higher dimensional categories I: Part 1*. Journal of Pure and Applied Algebra 153, pp. 221-246 (2000).
- [HMP01] C.Hermida, M.Makkai, J.Power. *On weak higher dimensional categories I: Part 2*. Journal of Pure and Applied Algebra 157, pp. 247-277 (2001).
- [HMP02] C.Hermida, M.Makkai, J.Power. *On weak higher dimensional categories I: Part 3*. Journal of Pure and Applied Algebra 166, pp. 83-104 (2002).
- [KM94] M.Kontsevich, Yu.Manin. *Gromov–Witten classes, quantum cohomology, and enumerative geometry*. Comm. Math. Phys., 164:3, pp. 525-562 (1994).
- [KS74] G.M.Kelly, R.Street. *Review of the elements of 2-categories* in *Category Seminar* Lecture Notes in Mathematics 420, pp. 75-103 Springer (1974).
- [Le04a] T.Leinster. *Operads in higher-dimensional category theory*. Theory and Applications of Categories, Vol. 12, No. 3, pp. 73-194 (2004).
- [MT00] H.Miyoshi, T.Tsujishita. *Weak ω -categories as ω -hypergraphs*. arXiv:mathCT/0003137v1
- [Pe99] J.Penon. *Approche polygraphique des ∞ -catégories non strictes*. Cahiers de Topologie et Géométrie Différentielle Catégoriques, 40 no. 1, pp. 31-80 (1999).
- [Str03] R.Street. *Weak omega-categories*. In *Diagrammatic morphisms and applications*. (San Francisco, CA, 2000), pp. 207-213, Contemp. Math., 318, AMS (2003).